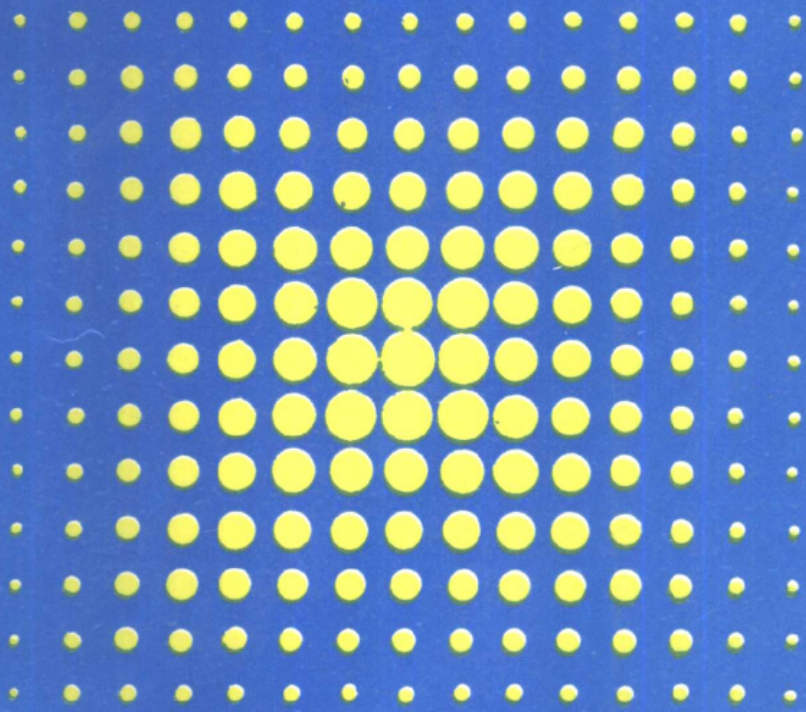
过程计算机控制丛书

过程仿真

何衍庆 编著



中国石化出版社

责任编辑　丁纵宇
封面设计　况　晗
责任校对　赫　青

ISBN 7-80043-596-2



9 787800 435966 >

ISBN 7-80043-596-2/TP·017
定价：23.00元

TP15
2120

过程计算机控制丛书

过 程 仿 真

何衍庆 编著

中国石化出版社

内 容 提 要

本书系统全面地介绍了计算机对各类控制系统的数字仿真问题及相应的控制策略、控制算法和仿真程序。

全书共分七章，包括连续系统、离散系统、采样系统、非线性系统的数字仿真及频域分析、最优控制和离散事件系统数字仿真等内容，通用性和实用性较强。

本书可供从事工业自动控制工程研究的科研人员、工程技术人员阅读，也可作为工业自动控制专业学生的参考教材。

图书在版编目（CIP）数据

过程仿真/何衍庆编著．－北京：中国石化出版社，1996（过程计算机控制丛书）
ISBN 7－80043－596－2

Ⅰ．过… Ⅱ．何… Ⅲ．过程控制－计算机仿真－数字仿真Ⅳ．TP273

中国版本图书馆 CIP 数据核字（95）第 15430 号

中国石化出版社出版发行
（北京朝阳区太阳宫路甲 1 号　邮政编码：100029）
海丰印刷厂排版印刷
新华书店北京发行所经销

787×1092 毫米 16 开本 $18^{1}/_{2}$ 印张　473 千字　印 1－3000
1996 年 4 月北京第 1 版　1996 年 4 月北京第 1 次印刷
定价：23.00 元

序

随着科学技术的发展，计算机在控制领域的应用越来越广泛。DCS（集散控制装置）、数字调节器、可编程控制器（PLC）等在工业自动化中的应用已迅速发展。与之同时，计算机在控制系统设计、研究与教学训练中的应用也日趋普遍。对这三类应用来说，计算机仿真是一门关键技术。

与实物模型实验相比，系统仿真有着许多明显的优点。例如，对象与控制器的结构和参数可以方便地自由变更，时间标尺可以自由选择，工作点的变化范围可以自由调整，用一台计算机可以对不同类型的物理系统进行仿真等。当然，实物模型实验也仍有其必要性，特别是内在机理不清晰、行为和特性不清楚的对象，是难以进行仿真的。

由于仿真手段的高度灵活性与通用性，现代的设计、研究与教学训练中几乎没有不采用仿真技术的。大家可能回忆起，30 年前是用模拟电路来构成仿真系统的，随着数字计算机的出现，模拟与数字混合型系统被认为是更好的方式，到了今天，绝大多数已被数字计算机仿真所代替。数字计算机仿真的最大特色是系统的不同结构、不同特性与不同参数都是通过软件来设置的，不需要改变具体的线路联接，因此，具有更大的灵活性与通用性。

一本介绍数字仿真技术的书的重心应该放在什么地方，这随书的目的而异。本书讨论的是各类控制系统的仿真，因此将控制系统的算法与分析的仿真作为重点，对各类控制系统、各种控制策略、各种分析方法作了比较系统和深入的介绍。因此，本书除了使读者熟悉相应的仿真方法以外，也使读者对现在常用的一些算法和分析方法有更多的了解。本书已注意和吸收了一些新的控制算法和策略。

近年以来，在计算机人机交互方面已有了很多改进。人们希望在仿真时操作更方便，显示更生动清晰，因此常常采用菜单、提示和各种窗口技术，在数据库和管理程序等方面也有许多讲究。这些技术具有一般性，不仅可用于仿真，也可用于其他方面。本书因篇幅限制，对这些很少涉及。好在今天这些方面的专门书籍已有不少，专用软件也很多。

本书的另一个特点是精选了相当数量的例题，精选了相当数量的程序，以便于读者的理解与应用。为使广大读者都能掌握与了解，这些程序采用 BASIC 语言编程。当然，按照不同的用途，采用 C 语言或其他高级语言也可能是合适的选择。

本书作者何衍庆副教授是一位基础深厚、工作细致、勤于事业的中年教师，他在理论分析、实验与仿真等方面都有很好的素养，对 DCS 的软件也很熟悉。为了写好这本书，他化费了大量的精力。他的这些优点，读者在本书的字里行间是不难觉察的。

为促进先进控制技术在过程控制领域的发展，我们组织了一套丛书，本书是其中之一。希望本书的出版有利于先进控制系统和策略的普及和推广。同时，也恳切希望读者对本书提出批评与建议。

蒋慰孙

1993 年夏写于华东理工大学自动化所

ABD02/10 02

前　言

本书是计算机应用丛书之一。重点介绍各类控制系统的数字仿真，对相应的控制策略和分析方法也作了一定的介绍和分析。

本书共分七章。第一章概述。第二章和第三章是连续系统的数字仿真，介绍了用离散化求解连续系统数值解的各种方法，第三章介绍了非线性系统的数字仿真。第四章是采样控制系统的数字仿真，介绍 z 域设计法和含纯滞后的采样控制系统的各种补偿控制方案。第五章是系统的频域分析，主要介绍衰减频率特性的分析方法和多变量逆奈奎斯特阵列法。第六章是最优控制系统的数字仿真，它是控制系统数字仿真和优化算法的结合，因此，该章主要介绍了各种优化算法。第七章是离散事件系统的数字仿真，介绍了随机数的产生和变换，以及离散事件系统的数字仿真。

本书的通用性和实用性较强。针对工业生产过程普遍存在的纯滞后过程，本书作了较为详细的分析和讨论。书中还附有大量的数字仿真程序、算法和实例，介绍了它们的使用方法和改进的措施。所附程序全部用 BASIC 语言编制，因为这种语言是初学者的计算机编程语言，初学者可以比较容易从中了解程序的设计思想。因此像窗口技术、菜单技术、绘图技术等较先进的计算机显示技术都较少涉及。有兴趣的读者可根据程序的设计思想和有关计算机技术结合自行编制。本书既可作为从事工业自动控制工程研究的科研人员、工程技术人员的技术参考书，也可作为工业自动控制专业学生的教材参考书。

蒋慰孙教授、俞金寿教授十分关心本书的编写工作，提出了不少宝贵的意见和建议。蒋慰孙教授还在百忙中抽时间仔细地审阅了全书文稿，详细地提出了修改意见。本书的编写和出版工作还得到了范秀兰、华向明、钱锋、钱幸生、陈杰、陈积玉、张美琴、严镇邦、胡长顺等同志的大力帮助和支持。谨在此一并表示衷心的感谢和深切的谢意。

由于时间和编著者水平有限，错漏在所难免，恳请读者不吝指正。

编著者

1993 年 6 月于华东理工大学

目　录

1　概述 …… 1

1.1 数字仿真的基本概念 …… 1
1.1.1 模型的分类及建模 …… 1
1.1.2 仿真的分类及仿真的基本方法 …… 2
1.1.3 数字仿真的基本特征 …… 3
1.2 数字仿真的基本原理 …… 4
1.2.1 相似性原理 …… 4
1.2.2 控制理论 …… 5
1.3 计算机仿真的应用实例 …… 5
1.3.1 精馏塔精馏过程仿真 …… 5
1.3.2 人口增长过程仿真 …… 6
1.3.3 北京地铁仿真模型及仿真结果 …… 6
1.4 仿真软件 …… 7
1.4.1 仿真软件 …… 7
1.4.2 仿真语言的性能评价 …… 7
1.4.3 典型数字仿真语言 …… 7

2　基于微分方程求解的连续系统数字仿真 …… 9

2.1 连续系统的数学模型 …… 9
2.1.1 数学模型的建立 …… 9
2.1.2 数学模型的类型 …… 11
2.1.3 数学模型的转换 …… 12
2.1.4 数学模型的合成 …… 16
2.1.5 纯滞后系统的数学模型 …… 18
2.2 数值积分算法 …… 19
2.2.1 数值积分算法的基本思想 …… 19
2.2.2 数值积分算法的一些基本概念 …… 21
2.2.3 数值积分算法 …… 22
2.2.4 算法的计算稳定性 …… 27
2.3 数字仿真程序 …… 29
2.3.1 数字仿真程序的基本结构 …… 29
2.3.2 数值积分算法子程序 …… 30
2.3.3 输入子程序 …… 33
2.3.4 微分方程组子程序 …… 35
2.3.5 主程序及输出子程序段 …… 36
2.3.6 纯滞后子程序 …… 37
2.3.7 数值仿真程序举例 …… 38
2.4 CSS 数值仿真程序包 …… 51
2.4.1 CSS 程序包的基本功能 …… 52
2.4.2 CSS 程序包清单及变量、数组说明 …… 54
2.4.3 CSS 仿真程序包的应用 …… 70

3　基于离散化模型的连续系统数字仿真 …… 76

3.1 连续系统的离散化模型 …… 76
3.1.1 采样定理 …… 76
3.1.2 离散化模型 …… 78
3.2 离散化模型的特性分析 …… 78
3.2.1 稳定性 …… 78
3.2.2 保持器 …… 79
3.3 离散化的方法 …… 82
3.3.1 z 变换法 …… 82
3.3.2 替换法 …… 86
3.3.3 根匹配法 …… 87
3.3.4 微分映射法 …… 91
3.3.5 状态变量法 …… 92
3.3.6 增广矩阵法 …… 96
3.4 非线性系统的数字仿真 …… 99
3.4.1 非线性环节的类型 …… 99
3.4.2 典型非线性环节的数字仿真子程序 …… 100
3.5 离散化模型的数字仿真程序 … 102

3.5.1 离散化模型的合成 …… 102
3.5.2 双线性变换仿真程序 …… 103
3.5.3 非线性系统数字仿真程序 …… 107

4 采样控制系统的数字仿真 …… 121

4.1 采样控制系统 …… 121
4.1.1 概述 …… 121
4.1.2 改进 z 变换 …… 122
4.1.3 数字控制器的描述 …… 124
4.2 采样控制系统的 z 域设计法 …… 127
4.2.1 差拍控制算法 …… 127
4.2.2 大林控制算法 …… 132
4.2.3 卡尔曼控制算法 …… 134
4.2.4 V.E. 控制算法 …… 136
4.3 含有纯滞后的采样控制系统 …… 138
4.3.1 史密斯预估补偿器 …… 138
4.3.2 观测补偿器 …… 141
4.3.3 解析预估补偿器 …… 144
4.4 采样周期 …… 146
4.4.1 不同采样周期对模型参数的影响 …… 146
4.4.2 不同采样周期时系统稳定性分析 …… 148
4.4.3 采样周期的选择 …… 151
4.5 采样控制系统数字仿真程序 …… 152
4.5.1 一阶滞后环节的数字仿真程序 …… 152
4.5.2 二阶滞后环节的数字仿真程序 …… 156
4.5.3 史密斯预估补偿控制数字仿真程序 …… 156

5 连续系统频域分析法的数字仿真 …… 162

5.1 频率特性 …… 162
5.1.1 典型开环传递函数及其频率特性 …… 162
5.1.2 频率特性的分析方法 …… 164
5.1.3 控制器参数整定 …… 166
5.2 衰减频率特性 …… 167
5.2.1 衰减频率特性 …… 167
5.2.2 衰减比判据 …… 168
5.2.3 用衰减频率特性整定控制器参数 …… 169
5.2.4 衰减频率特性仿真程序 …… 173
5.3 劳斯判据确定系统稳定性 …… 178
5.3.1 劳斯稳定判据 …… 178
5.3.2 劳斯判据确定系统相对稳定性 …… 179
5.3.3 仿真程序 …… 180
5.4 多变量频域的逆奈氏阵列法 …… 187
5.4.1 基本概念 …… 187
5.4.2 多变量闭环系统的稳定性 …… 189
5.4.3 补偿器的设计 …… 191

6 最优控制系统的数字仿真 …… 197

6.1 概述 …… 197
6.1.1 目标函数和约束条件 …… 197
6.1.2 最优控制系统的数学描述 …… 198
6.1.3 目标函数取极值的充要条件 …… 199
6.2 静态最优化方法 …… 200
6.2.1 线性规划方法 …… 200
6.2.2 爬山法 …… 207
6.2.3 优选法 …… 213
6.2.4 单纯形加速法 …… 214
6.2.5 拉格朗日乘子法 …… 224
6.2.6 静态优化的实例 …… 231
6.3 动态最优化方法 …… 234
6.3.1 变分法 …… 234
6.3.2 极大值原理 …… 239

6.3.3 动态规划法 …… 255

7 离散事件系统的数字仿真 …… 258

7.1 基本概念 …… 258
7.1.1 随机变量 …… 258
7.1.2 随机过程 …… 260
7.2 随机数的产生和变换 …… 261
7.2.1 伪随机数的产生 …… 261
7.2.2 随机变量的产生 …… 262
7.2.3 多维随机变量的产生 …… 268
7.2.4 随机过程的产生 …… 269
7.2.5 数字仿真子程序 …… 271
7.3 离散事件系统的数字仿真 …… 274
7.3.1 离散事件系统的数学模型 …… 274
7.3.2 离散事件系统的仿真策略和结构模型 …… 277
7.3.3 离散事件系统仿真结果分析 …… 285
7.3.4 方差减小技术 …… 287
7.3.5 通用离散事件系统仿真语言 …… 288
参考文献 …… 289

1 概　述

1.1 数字仿真的基本概念

1.1.1 模型的分类及建模

数字仿真可广义地理解为对真实客体的形态、工作规律或信息传递规律在特定条件下用数学的形式进行描述的相似性复现。通过模型的运行来实现对实体的数字仿真是藉助于数字计算机实现的。

模型是把所研究的实体作为一系列相互关联的部分所构成，各部分在整体结构中发挥一定的作用，用数学的方法来对实体进行相似性的复现。因此，建模是进行数字仿真的前提，而通过数字仿真又可以检验模型、修正模型，使之能比较真实地反映实体的特征。

从哲学的观点来分，模型可分为两大类。第一类是物质模型，它们是不以人们的意志为转移的客观存在的实体。几何模型是再现实体原型的空间形式的物质模型。例如三角形就是表示具有一定属性的几何模型。物理模型除了要求与原型空间相似外，还要求它们的物理特性相似。精馏塔、反应器及其他设备的模型都属于物理模型。数学模型则是从一定的功能或结构上进行相似，用数学的方法来再现原型的功能或结构特征。可以把真实系统看作是产生一定性状数据的一个源。而数学模型则是一些规则、指令的集合，它产生与真空系统相同的性状数据。例如，一个控制系统可以用状态方程描述的数学模型来分析。

第二类是观念模型，它是客观存在的非物质因素为原型所建立的模型。仿生学的研究使人们从鸟的翅膀形成对飞机制造的肖像模型、数理逻辑中应用的命题表达、命题演算等用符号或逻辑关系描述思维过程的符号模型及化学理论中用原子价、化学符号等符号元素描述分子结构特征的混合模型等都属于观念模型。

也可以从不同角度对模型进行分类。

按模型中时间集的取值，可分为连续时间模型——其时间集取全体实数集或实轴上的区间；离散时间模型——其时间集取固定时间整数倍的序列。按模型中描述变量之间的函数关系是否依赖于时间，可分为时不变模型——即函数关系中只包含描述变量，不包含时间变量；时变模型——函数关系中，还把时间作为自变量。

根据模型中描述变量的轨迹（在一定时间内的值的变化），可分为连续模型——它的状态变化是连续的，状态的变化率满足一定的微分方程；离散事件模型——它的时间变化虽然是连续的，但状态变化表现为在各个离散时刻的跳跃，即状态变化以事件形式出现在各个离散的、可取任意值的时刻。当模型中的描述变量不包含随机变量时，称为确定性模型；包含随机变量时，称为随机模型。

考虑外界对实际系统是否有影响，模型可分为自治模型——不考虑输入变量，即外界影响；非自治模型——包含输入变量，考虑外界对系统的影响。输入和输出变量均不包含的模型称为封闭模型；反之，包含输入或输出变量，或二者均包含的模型，称为开放模型。

建立模型应遵循下列原则：

①精确性　精确性是模型与真实客体的相似程度的标志，根据所研究问题的目的、内容的不同对模型的精确度有不同的要求。建模的方法、测试数据的精确度、模型的结构都对模型的精确度造成影响。因此，应从工程实际要求出发，确定合适的模型精确度。

②合理性　模型是对被研究实体在特定条件下的相似性复现。因此，在模型建立前，合理地提出模型适用条件是十分必要的。例如，建立换热器的模型时，若被加热物料和加热物料都是均匀搅拌的，那么，建立的模型就是集中参数模型。若随着物料的移动，物料温度随着距离而变化，则建立的模型是分布参数模型。

③复杂性　在所需模型精确度的前提下，应对模型进行合理的简化。降低模型的阶数和简化模型结构是降低模型复杂程度的常用方法。为了研究的需要，对被研究的实际系统进行分解是必要的，但不应认为分得越细越好。相反，在一些场合，通过必要的简化，不仅可以使程序简单，而且也节省了大量的仿真时间。

④应用性　建立模型的目的是通过对模型的仿真，应用于实际系统的分析和研究。因此，对模型中描述变量的选择应该从实际应用出发，遵循输入变量可测性准则。

⑤鲁棒性　模型的适应性不仅与模型的精确性有关，而且还与模型的结构、参数等有关。一个鲁棒性好的模型不仅在所假设的工况条件下是适用的，而且当工况条件在假设条件以外的一定范围内也是适用的。这对于仿真结果的外推十分必要。

模型的建立有各种方法。常用的方法是机理模型、回归模型或两者的结合方法。

机理模型的建立是通过对系统的机理分析、演绎而获得的。回归模型的建立是通过对系统输入输出数据的分析、回归而得到的。通过机理分析及实测数据的回归确定机理模型中的参数，这种方法是混合建模的方法。

1.1.2　仿真的分类及仿真的基本方法

根据模型的不同，仿真的方法也不同。

几何模型是以几何相似为其出发点的，几何仿真的研究实际上是几何学研究的课题。例如，通过三角形的几何模型，可以获得三边中线交于一点，此点是三角形的重心的结论。

物理模型以物理系统为模型。通过对该物理模型的仿真可以获得与该物理系统有相似物理特性的系统的仿真结果。风洞试验就是对飞机的空气动力学特性的物理仿真。通过对用运算放大器组成的一阶惯性环节的物理仿真来研究液位贮槽中进料量与液位的关系则是应用了物理特性的相似性进行仿真的例子。

数字仿真是对系统的数学模型进行研究，模拟实际系统中变量的变化，获取系统输出变量特性的仿真方法。对于简单的数学模型，通过解析的方法可以直接进行研究，但对复杂的系统，通常是通过计算机对数学模型进行研究和试验的。因此，计算机数字仿真常简称为数字仿真。

在有些系统的研究中，把计算机和实际的一些物理系统相联接，把数字仿真和物理仿真结合起来，这种仿真称为混合仿真，或称数学——物理仿真。

根据被仿真系统的不同，仿真的类型可以分为连续系统仿真、离散系统仿真、采样系统仿真、分布参数系统仿真、离散事件系统仿真等。

随着数字仿真内涵的扩展，根据数字仿真的任务不仅包含系统的分析、研究，还包括系统的设计和操作人员的培训仿真器。因此，一些计算机辅助设计的内容及仿真语言的开发等也越来越为数字仿真的研究者所重视。数字仿真的内容已涉及到下列各方面：

①仿真计算机和仿真系统　模拟计算机、数字计算机、模拟——数字混合计算机、微处

理机阵列机、并行多处理机系统。

②仿真方法　模拟仿真方法、数字仿真方法、模拟——数字混合仿真方法、离散事件系统仿真方法、蒙特卡罗（Monte－Carlo）法。

③仿真程序和仿真语言　各种类型的仿真程序和仿真语言、图形仿真、仿真数据库。

④仿真研究　多次运行、参数寻优、函数寻优。

⑤辅助设计　单变量、多变量系统的辅助设计、时域和频域的辅助设计、最优化设计。

⑥仿真器　用于各种应用领域的训练、培训用仿真器。

⑦仿真基准问题　生物学基准、科学混合基准。

系统仿真是根据系统的数学模型，转化为计算机可以实施计算的仿真模型，并在仿真模型上进行分析、研究。仿真的基本方法是建立仿真模型和编程、仿真运算。可用图1－1表示数字仿真的基本步骤。图中，虚线框内的部分就是数字仿真的内容。

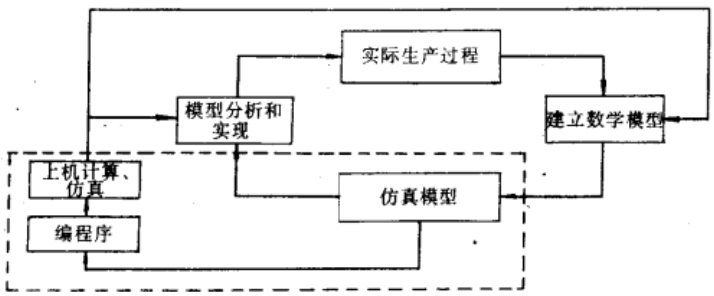


图1－1　数字仿真步骤

1.1.3　数字仿真的基本特征

数字仿真是一门数值计算技术。它包含了计算数学、计算机技术、控制论、概率统计学等方面的知识，它以过程辨识和建模为基础，用于对过程的仿真模型进行分析、研究和设计。因此，数字仿真技术是一门新兴的综合性技术课程。

数字仿真的基本特征是：

①对于真实客体的研究，往往由于它的复杂性，难于直接进行研究，为了深入地对这种过程进行研究、提出改进方案或者作出正确评价和论证，常常要化费大量的人力、物力，财力和时间。而且对于有些过程实体的研究，会因为有危险或不易实施某些条件而无法进行。采用计算机进行数字仿真就可以避免上述问题，可以通过对仿真结果的分析，作出正确判断，并能利用系统现有信息，预测未来，为系统的改进、利用提供有用信息。

②对于新系统、新方案的设计，可以不必急于无根据地花费大量投资去实际实施，而可以通过计算机仿真，对新系统、新方案进行分析、研究，作出正确评价和论证。

③计算机仿真可以应用于自然科学和技术的研究、分析，也可以应用于社会科学等领域的研究，分析。计算机仿真技术的出现将改变人们的观念，改变社会生产和生活方式，使各学科间相互交汇，推动社会进步。

④在控制和决策时，可以利用计算机对具有不同控制变量、决策变量、状态变量及有关参数进行多种方案仿真，制订出最优控制和决策。

⑤通过数字仿真，可以在一个较长时间内对系统进行研究，时间和被考虑的因素由于计算机的大容量存贮和高速度运算而不必简化，从而发现系统的有用信息。而且通过系统运行规律的分析、研究，还可以预测未来系统可能的运行轨线。

⑥对于系统中含有的不确定因素、随机因素可以利用计算机仿真方法产生随机数，利用统计方法和模糊数学的有关规则对系统进行仿真。

⑦计算机仿真只能提供系统在某一特定条件下的特解，不能象解析分析方法那样可以得到解析解。此外，为了寻优，需在计算机上进行大量的重复运算，才能按一定规则找到可行解中的最优解。

⑧为了确定系统的初始运行条件，常常要收集大量运行数据，分析比较，花费大量时间，占用计算机内存和影响仿真速度。

1.2 数字仿真的基本原理

1.2.1 相似性原理

相似性的概念最早出现在几何学里，两个三角形的相似是指这两个三角形的对应边成比例，或者对应角相等。这种相似，称为形状的几何相似。当我们对同一类现象中的某一种现象进行研究，获得了有关信息和规律，并应用于这类现象的其他任一种现象时，这种相似称为同类相似。例如，通过对某一典型的换热器的研究和分析，得到的逆流换热的有关规律可以应用于类似的逆流换热的换热器分析。对于两类不同现象的质的相似性研究属于异类相似。例如传热对象、液位对象和 RC 电路中的时间常数都是容量系数和阻力系数之积，平衡关系参数是容量系数和被控参数（推动力参数）之积，而流量系数则是推动力参数除以阻力。其关系如表 1－1。

表 1－1 热、液、电三类现象的相似关系

平衡关系参数	容量系数	被控参数（推动力参数）	流量参数	阻　力	时间常数
热量 u	热容 M_c	温度 T	热流 q	热阻 $\frac{1}{G_{c_0}\cdot C}$	$M_c\frac{1}{G_{c_0}\cdot C}$
体积 V	面积 F	液位 h	流量 Q	液阻 $\frac{2\sqrt{h_0}}{C_v}$	$F\cdot\frac{2\sqrt{h_0}}{C_v}$
电荷 e	电容 C	电压 u	电流 i	电阻 R	$C\cdot R$

根据相似关系，三类现象的输入流量参数和输出被控参数之间都可用传递函数 $\frac{K}{Ts+1}$ 描述。因此，对这三类不同种类现象的输入和输出关系的研究就可以用对一阶惯性环节的研究来类推。

相似性原理由相似性正定理、逆定理与推广定理三部分组成。

相似性正定理指出彼此相似的现象必定具有数值相同的相似准则。这表明相似现象的一切量各自互成比例。由这些量组成的方程组是相同的。

相似性逆定理指出了两类现象相似应满足的条件。这些条件是，可以用同一完整的方程组来描述；单值条件的物理量组成的相似准则在数值上相等。

相似性推广定理指出描述某现象的各种量之间的关系可以整理成相似准则关系式，则这些准则关系可以推广应用到与其相似的现象中去。正是基于这个原理，我们才能用数字仿真的方法把研究的结果应用于相似现象的分析研究中去。

相似性原理是仿真的基础。相似准则可以描述为几何的、物理的或数学的形式。数字仿真是用数学的描述形式来对相似现象从性能上进行的研究。

1.2.2 控制理论

对系统的控制是通过控制指令中所包含的信息得到的。即控制是和观察的信息、被控制系统和与它起作用的外部介质的信息及控制作用的结果信息等各方面联系在一起的。系统与介质的信息交换及系统内部各部分之间的信息交换都是藉助于各种类型的连接关系进行的。这种联系是控制论系统的特征。反馈就是这样的一种联系通道，有关控制结果的信息通过它送回到系统中去。

控制论是组织界系统的理论。它的重点是分析系统的行为和被控制对象的信息，分析信息一调节过程和系统行为、功能的性质。在美国，控制论的特点被看成是同研究、推广和使用计算机技术有关的种种问题的总和。控制论处处应用着数学方法，但它不是数学的一个分支。在研究控制论系统作用原理时，它脱离了对物质某种形式运动的从属性，而把研究对象局限于控制和信息的过程，从而揭示出被研究对象的特征。

按结构水平原理对控制论分类，则有社会控制论、生物控制论和技术控制论三类。近年来，根据三类控制论的组合，产生了军事控制论、医学控制论、经济控制论等。

技术控制论包括了机器装置和功能的理论和很多数学控制论的内容，例如，信息统计、自动调节和控制、算法语言、程序设计等等。

计算机仿真是按控制的理论和方法，首先把被研究客体诸量信息化，构造实质内容的模型；其次，在上述模型基础上，建立用数量的关系描述的仿真模型；然后，编制程序，使计算机按一定的规则对上述信息进行处理，并获得它们的信息关系和仿真结果。可见，计算机仿真的方法是控制论研究客体的具体体现。

总之，相似论、控制论及近代数学的有关学科的发展，不仅为计算机仿真提供了理论依据，而且为计算机仿真提供了方法和技巧。近代科学，如人工智能、预测技术、系统分析等科学的发展，将对计算机仿真提供更有效的方法和理论，使计算机仿真技术的应用更广泛更深入。

1.3 计算机仿真的应用实例

1.3.1 精馏塔精馏过程仿真

精馏过程是石油化工生产过程中最常见和最重要的单元操作过程。精馏过程的仿真包括动态数学模型建立和动态仿真，它是在静态数学模型基础上建立起来并经简化得到的模型，并据此进行的数字仿真研究。

精馏塔静态数学模型的建立有逐板法和三对角阵法，逐板法先对每一个塔板建立汽一液平衡方程，物料和能量的衡算方程式等基本方程式，然后，从塔顶开始，按各轻组分浓度向下逐板计算，直至啮合板（一般选进料板）。此外，从塔底开始，按塔底各重组分的浓度，向上逐板计算，直至啮合板，根据上下行计算所得的啮合差进行修正，直到满足要求为止。

建立动态数学模型是由于静态数学模型反映的是精馏塔在稳态时的各塔板参数，而动态数学模型可以得到当控制变量变化时对各塔板参数的影响，从而为优化控制方案的建立提供依据。此外，动态数学模型是为一定目的而经过简化的，它能为实时仿真提供可能性。

通过动态仿真，可以看到，对于多元物系的工业精馏塔，它的非线性特性极为明显，可用变参数的简化动态数学模型进行描述。实际应用表明动态仿真是正确的，已在多种类型精馏塔中得到应用。

1.3.2 人口增长过程仿真

人口的控制对于国民经济的发展有着密切的关系。设在 t 时刻的人口总数为 $N(t)$。在下列简化假设条件下，可列出人口模型：

①未来人口总数是现有人口的函数。

②性别比率及生育率为定值。

③不同年龄组人口间比例保持定值。

④无移民的变化及生活环境的突发变化。

⑤人口净增长率 λ 为常数。

人口模型可用下式描述：

$$\frac{dN(t)}{dt} = \lambda N(t)$$

当 $\lambda > 0$ 则人口会不断增长，达到某一警戒率。当 $\lambda = 0$ 则人口维持不增不减的状态，出生人数等于死亡人数。当 $\lambda < 0$ 则人口不断减少。在短期的人口预测中，这种简化模型是有效的。表 1－2 是某国从 1800 到 1900 年之间观察值与预测值的比较。

表 1－2 某国人口增长观察和仿真结果

年	观察值（百万）	仿真值（百万）
1800	5.3	5.3
1820	9.6	9.7
1840	17.1	17.4
1860	31.4	30.3
1880	50.2	50.1
1900	76	76.9

实际应用的人口模型要复杂得多。我国学者宋健等根据我国国情建立的人口模型为制订国民经济计划、人口政策等起了积极的作用。

1.3.3 北京地铁仿真模型及仿真结果

北京地铁早期工程东起北京火车站，西至苹果园，共 17 个站，全长 23.6km，双向运行，单向运行全程 32 分钟，各站停留共 8 分钟，根据客流调查所得的近 30 万个数据，建立了北京地铁的随机服务系统模型，以原始时刻表作为初始值进行列车发送，在动态仿真中，根据乘客平均等待时间及列车满载率对时刻表进行调整，最后得出了满足给定条件下的发车时刻表和相应的仿真结果。

这个系统是一个随机服务系统，采用离散事件系统仿真时是用 GPSS 语言实施的。通过仿真给出了客流方面的有关信息，如各站车上人数、上车、下车人数、进站人数、未能上车人数、乘客平均等待时间及列车运动的有关信息，如列车运行时刻表、运行图、各列车满载率、最大和平均满载率等等。由于采用计算机进行计算和仿真，大大缩短了时间，为合理编制客运计划，科学安排运输能力，提高管理水平发挥了极重要的作用。

计算机仿真在其他领域的应用实例可以举出很多，在工程技术、社会经济、商业管理等领域的应用使得实验成本下降、安全性提高，为优化决策提供了依据。此外，用于训练的仿真器已在航空、航海等运载工具的训练培训和化工、电站等工业设备的操作训练中得到应用。

1.4 仿 真 软 件

1.4.1 仿真软件

为了实施数字仿真，必须把仿真模型用计算机所能执行的语言进行编辑，使之能够执行。仿真软件是实施上述功能的应用软件。仿真软件可分为仿真程序、仿真程序包和仿真语言等。通常仿真程序是为解决某一类实际问题而编制的程序，它占用的内存较少，通用性较差，在数字仿真的初期或为解决某些特定问题时被广泛采用。仿真程序包则解决的面较广，如面向微分方程的连续系统数字仿真程序包能够对以微分方程形式描述的连续系统进行数字仿真，通常可以按程序包的使用说明直接使用，一般无需具有计算机编程的有关知识，但为了完成程序包的要求，需对程序包提供的各种模块有较深入的研究，才能组成所需要研究的仿真模型。仿真语言则有自己一整套的规定语句、符号、句法和语法，用户可根据被研究的仿真模型按仿真语言的有关语句编写结构参数、系统参数等，使用较方便，随着人工智能，专家系统的应用，会得到更广泛的应用。但对初学者来讲，由于他们对数字仿真，或者使用的应用软件缺乏实践，因此，应作必要的培训和指导。本书考虑到应用的环境、选用了初学者能自学的 BASIC 语言编程，并在国内有较大用户的 IBM 微机上实施了这些仿真程序。此外有软盘可供选用。

1.4.2 仿真语言的性能评价

由于仿真的问题不同，为适用于不同系统的数字仿真，对仿真语言也提出了适用性的问题。因此一般仿真语言都有若干种仿真算法可供用户选用。对仿真语言的性能评价虽与仿真语言的功能和设计要求有关，常可用下列性能进行评价。

①语言表达性　仿真语言对模型的表达能力。包括模型类型、符号程序库、数据处理等。

②数值能力　仿真语言对模型数值的处理能力。

③语言结构特性　仿真语言是否有利于建立模型、进行仿真实验、转换成机器码等。包括排序算法、分区描述及模块化等。

④程序执行性能　程序能否采用人机交互式执行，可否变换模型参数和系统参数、可否改变模型结构等。

⑤数据处理性能　对模型结构参数、初值、输入、输出数据存贮、检索和运算处理能力。

⑥输入、输出特性　数据和图形的输入和输出。

⑦模型鉴定　对数学模型的有效性和对仿真模型可信度的确定。

⑧语言实现方式　程序的实施运行有直接执行、解释型、编译型等。

⑨专家功能　有否人工智能、自学习等功能。

1.4.3 典型数字仿真语言

数字仿真语言在国外于 60 年代初就开始研制。美国计算机仿真协会软件委员会在 1967 年推出了数字仿真语言的标准文本 CSSL。而在此同时，由 IBM 公司推出了 CSMP 数字仿真语言。这些语言面向微分方程组，适用于连续系统的数字仿真。采用时间离散的连续系统数字仿真语言有 DYNAMO，即动力学模型。

离散型系统的数字仿真语言按工作原理可分为三类。以事件为基础的 SIMSCRIPT、

SIMLIB 等，以活动扫描为基础的 CSL 及以过程为基础的 GPSS 等。

既可用于连续系统又可用于离散系统的数字仿真语言有 GASP－Ⅳ，SLAM 等。此外尚有专用的仿真语言，如通用机械制造系统计算机仿真语言 GCMS 等。

随着人工智能科学的发展，出现了 PROLOG、LISP 语言，这种表处理的语言使人工智能在计算机上的实现成为可能。

我国在 70 年代初引进了一些数字仿真程序包和仿真语言，经消化、移植，推出了一些有特色的仿真语言与程序包，如 ZFX、DYNAMO、CSSP、SMPACK、ICSL 等，并得到了广泛的应用。

2 基于微分方程求解的连续系统数字仿真

用计算机进行数字仿真，首先应把系统的数学模型转化为计算机能够采用接受的仿真模型。对于不同的计算机仿真算法，所需的仿真模型是不同的。本章讨论的数字仿真采用数值积分仿真算法，所需的仿真模型是一阶微分方程或一阶微分方程组。

本章介绍实际控制系统中，连续系统的几类数学模型描述，以及它们转化为仿真模型的方法。讨论了数值积分的几种不同算法及其计算稳定性问题。最后，本章给出了用BASIC语言编制的实用数字仿真程序。

连续系统可分为不含纯滞后环节和含有纯滞后环节等两类。本章对这两类系统都进行了讨论。

2.1 连续系统的数学模型

2.1.1 数学模型的建立

系统动态特性的数学表达式，称为数学模型。分析研究动态系统，首先应建立系统的数学模型。工业生产过程中，控制系统的数学模型可通过对生产过程机理的分析，应用物理、化学等有关的定律和知识，经推导获得。由于工业生产过程的复杂性，大多数系统的数学模型则是通过对生产过程的辨识获得的。也就是在生产过程的输入端加入某种类型的激励信号，在输出端测取系统的响应输出信号，通过某种系统辨识的方法确定所测系统的结构及相应参数。工业生产过程中常用的测试方法是响应曲线法。控制系统的数学模型常见的有三种形式。它们是高阶微分方程（微分算子方程）、状态空间（一阶微分方程组）及传递函数。下面以实例说明之。

〔例2－1〕连通系统的数学模型

图2－1所示的三个连通容器，已知其容器横截面积分别为 S_1、S_2、S_3，液位分别为 H_1、H_2、H_3，各阀阻力系数分别为 k_1、k_2、k_3。试求液面 H_3 与进料量 Q 之间的关系。

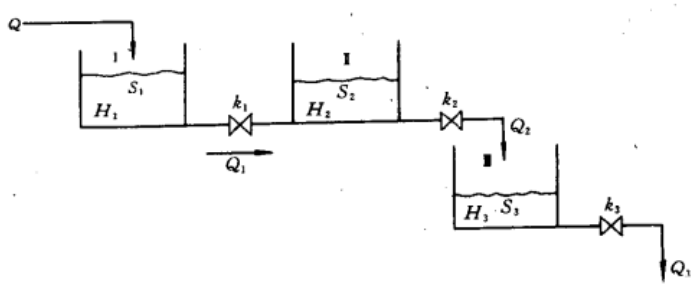


图 2－1　连通系统

根据流体力学有关知识，有：

$$Q_i^2 = k_i \Delta P_i \qquad (i = 1,2,3)$$

对容器Ⅰ有：$S_1 \dfrac{dH_1}{dt} = Q - Q_1$

对容器Ⅱ有：$S_2\frac{dH_2}{dt}=Q_1-Q_2$

对容器Ⅲ有：$S_3\frac{dH_3}{dt}=Q_2-Q_3$

式中，H_1、H_2、H_3、Q_1、Q_2、Q_3、Q 均已增量化。根据静态条件，经线性化，得：

$$Q_1=\frac{k_1(H_1-H_2)}{2Q_{10}};Q_2=\frac{k_2H_2}{2Q_{20}};Q_3=\frac{k_3H_3}{2Q_{30}}$$

式中，Q_{10}、Q_{20}、Q_{30}是稳态时相应流量。

化简得：

$$\frac{d^3H_3(t)}{dt^3}+A_1\frac{d^2H_3(t)}{dt^3}+A_2\frac{dH_3(t)}{dt}+A_3H_3(t)=A_3Q(t) \qquad (2-1)$$

式中，$A_1=\frac{2Q_{10}}{k_1}$；$A_2=\frac{2Q_{20}}{k_2}$；$A_3=\frac{2Q_{30}}{k_3}$；

式（2－1）是用三阶微分方程描述液位 H_3 与进料量 Q 之间关系的数学模型。

〔例2－2〕换热器系统

为测取图2－2所示的换热器系统控制阀开度和物料出口温度之间的关系，在系统开环时，控制阀开度从50%阶跃变化到60%，测得物料出口温度的响应曲线如图2－3所示。

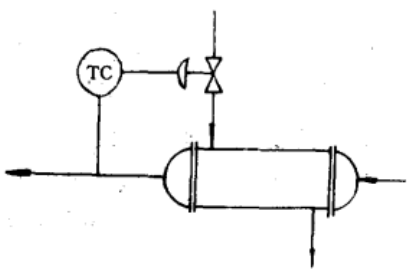


图2－2　换热器系统

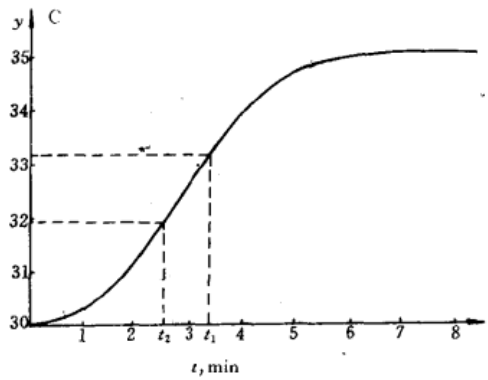


图2－3　响应曲线

设所用温度变送器量程0～50℃，系统的数学模型用一阶加纯滞后环节，即 $G(s)=\frac{K}{Ts+1}e^{-\tau s}$ 近似。根据系统辨识的有关知识，可求得：

$$K=\frac{\frac{35-30}{50-0}}{\frac{60-50}{100-0}}=1$$

时间常数 T 与纯滞后 τ 可用切线法或两点法求得。采用两点法时，在响应曲线上取两点，

$$\begin{cases} C(t_1)=0.632 & t_1=3.4\text{min} \\ C(t_2)=0.384 & t_2=2.55\text{min} \end{cases}$$

$$\because \quad C(t)=K(1-e^{-\frac{t-\tau}{T}})$$

$$\therefore \quad T=\frac{t_1-t_2}{\ln(1-C(t_2))-\ln(1-C(t_1))}=1.65\text{min}$$

$$\tau = t_1 - T = 1.75\text{min}$$

系统的数学模型 $G(s)$ 可近似表示为

$$G(s) = \frac{e^{-1.75}}{1.65s + 1} \tag{2-2}$$

式（2－2）是用传递函数描述换热器系统中物料出口温度和进料控制阀开度之间关系的数学模型。

〔例 2－3〕 *RLC* 电路.

图 2－4 是典型的 *RLC* 电路，为研究电路输入电压 V 与输出电压 V_C 的关系，根据基尔霍夫定律，可列出状态方程

$$\begin{bmatrix} \dot{i}(t) \\ \dot{V}_C(t) \end{bmatrix} = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{L} \\ \frac{1}{C} & 0 \end{bmatrix} \begin{bmatrix} i(t) \\ V_C(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} V(t) \tag{2-3}$$

及输出方程

$$V_C(t) = V_C(t) = [0 \quad 1] \begin{bmatrix} i(t) \\ V_C(t) \end{bmatrix}$$

式（2－3）是用状态空间描述 *RLC* 电路中状态变量 $i(t)$、$V_C(t)$ 与输入变量 $V(t)$、输出变量 $V_C(t)$ 之间关系的数学模型。

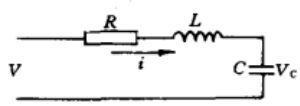


图 2－4 *RLC* 电路

可以看到，有些数学模型中含有纯滞后项，有些数学模型则不含有。在化工生产过程中，由于存在物料的传递、传质和传热过程，或多或少地存在纯滞后现象，因此研究含有纯滞后的连续系统，对这类系统进行数字仿真尤为必要。

2.1.2 数字模型的类型

连续控制系统的数学模型，有高阶微分方程、传递函数和状态空间表达式等，在一般控制理论书籍中都有详细介绍，这里仅作简单说明。

2.1.2.1 微分方程

设连续系统的输入是量为 $u(t)$，输出量为 $y(t)$，则一个连续控制系统可用下面的 n 阶微分方程来描述

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \cdots a_{n-1} \frac{dy}{dt} + a_n y = c_0 \frac{d^n u}{dt^n} + c_1 \frac{d^{n-1} u}{dt^{n-1}} + \cdots + c_{n-1} \frac{du}{dt} + c_n u \tag{2-4}$$

式中系数 a_1、a_2、…、a_n 与 c_0、c_1、…、c_n 若为常数，则称为常微分方程，若与时间 t 有关，则称为时变的，例 2－1 是用三阶微分方程描述连通系统的例子。

2.1.2.2 传递函数

对控制工程师来讲，他们更习惯用传递函数来描述控制系统。若系统的输入 u 与输出 y 的各阶导数和初值都为零，则对（2－4）式等号两边逐项进行拉氏变换，可得系统的传递函数为：

$$\frac{Y(s)}{U(s)} = \frac{c_0 s^n + c_1 s^{n-1} + \cdots + c_{n-1} s + c_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \tag{2-5}$$

式中，$Y(s)$是输出的拉氏变换，$U(s)$是输入的拉氏变换。

2.1.2.3 状态空间

n阶控制系统的状态空间表达式是：

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2-6}$$

$$y(t) = Cx(t) + Du(t) \tag{2-7}$$

式中，A是$n \times n$维系数矩阵。B是$n \times p$维输入系数矩阵，p是输入变量的个数。(2-7)式称输出方程，(2-6)式称状态方程，合称状态空间表达式。它实际上是一阶微分方程组的矩阵形式的描述。

2.1.3 数学模型的转换

为了便于用计算机对连续系统进行数字仿真，当采用数值积分算法时，系统的数学模型要转换成计算机可应用的仿真模型，即以一阶微分方程组（或状态空间表达式）形式描述的仿真模型。状态空间表达式描述的数学模型符合仿真模型的要求，不需转换外，以高阶微分方程或传递函数形式描述的数学模型都要转换成仿真模型。如前述，高阶微分方程可经拉氏变换转换为传递函数，因此，数学模型的转换主要讨论传递函数转换到状态空间表达式的问题。

根据现代控制理论，选用不同的状态变量，同一个系统可以转换成不同形式的状态方程和输出方程。选择合适的状态变量可以简化数字仿真的计算。因此，这里介绍几种由传递函数转换为状态空间表达式的方法。对于多输入多输出的系统，其转换方法可参考现代控制理论的有关书籍。

2.1.3.1 直接分解法（Direct Resolution）

作为一般形式的传递函数可描述如下：

$$G(s) = \frac{Y(s)}{U(s)} = \frac{c_0 s^n + c_1 s^{n-1} + \cdots + c_{n-1} s + c_n}{a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n} \tag{2-8}$$

引入变量$Z(s)$：

$$Z(s) = \frac{\frac{1}{a_0} U(s)}{1 + \frac{a_1}{a_0} s^{-1} + \cdots + \frac{a_{n-1}}{a_0} s^{-n+1} + \frac{a_n}{a_0} s^{-n}}$$

及状态变量$x_i(s)$：

$$x_i(s) = s^{-n+i-1} Z(s) \qquad (i = 1, 2, \cdots, n)$$

则有：

$$\begin{cases} \dot{x}_i = x_{i+1} & (i = 1, 2, \cdots n-1) \\ \dot{x}_n = Z = \frac{1}{a_0} u - \frac{a_1}{a_0} x_n - \cdots - \frac{a_{n-1}}{a_0} x_2 - \frac{a_n}{a_0} x_1 \\ y = c_0 Z + c_1 x_n + \cdots + c_{n-1} x_2 + c_n x_1 \end{cases} \tag{2-9}$$

用状态空间的标准形式，可描述如下：

$$\begin{cases}\dot{\boldsymbol{x}}=\begin{bmatrix}0 & & & \\ \vdots & & I_{n-1} & \\ 0 & & & \\ -\dfrac{a_n}{a_0} & -\dfrac{a_{n-1}}{a_0} & \cdots & -\dfrac{a_2}{a_0} & -\dfrac{a_1}{a_0}\end{bmatrix}\boldsymbol{x}+\begin{bmatrix}0\\ \vdots\\ 0\\ \dfrac{1}{a_0}\end{bmatrix}u & (2-10)\\ y=\left(c_n-\dfrac{c_0}{a_0}a_n,\ c_{n-1}-\dfrac{c_0}{a_0}a_{n-1},\ \cdots,\ c_1-\dfrac{c_0}{a_0}a_1\right)\boldsymbol{x}+\dfrac{c_0}{a_0}u & (2-11)\end{cases}$$

直接分解法的状态变量图如图 2－5 所示。

当 $c_0=0$，系统是严格正则的（Strict Proper）。这时输出方程成为：

$$y=[c_n,c_{n-1},\cdots,c_1]\boldsymbol{x} \quad (2-12)$$

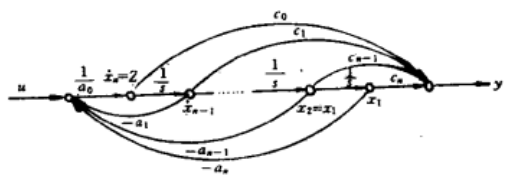


图 2－5　直接分解法

2.1.3.2　交叉分解法（Cross Resolution）

若把（2－8）式改写成：

$$Y(s)=\frac{c_0}{a_0}U(s)+\frac{1}{s}\left[\frac{c_1}{a_0}U(s)-\frac{a_1}{a_0}Y(s)\right]+\cdots+$$
$$+\frac{1}{s^n}\left[\frac{c_n}{a_0}U(s)-\frac{a_n}{a_0}Y(s)\right]$$

令

$$\begin{cases}\dot{x}_i=x_{i+1}+\dfrac{c_i}{a_0}u-\dfrac{c_i}{a_0}y & (i=1,2,\cdots,n-1)\\ \dot{x}_n=\dfrac{c_n}{a_0}u-\dfrac{a_n}{a_0}y & (2-13)\\ y=x_1+\dfrac{c_0}{a_0}u\end{cases}$$

其标准形式如下：

$$\dot{\boldsymbol{x}}=\begin{bmatrix}-\dfrac{a_1}{a_0} & & & \\ -\dfrac{a_2}{a_0} & & & \\ \vdots & & I_{n-1} & \\ -\dfrac{a_{n-1}}{a_0} & & & \\ -\dfrac{a_n}{a_0} & 0 & 0\ \cdots\ 0\end{bmatrix}\boldsymbol{x}+\begin{bmatrix}\dfrac{a_0c_1-a_1c_0}{a_0^2}\\ \dfrac{a_0c_2-a_2c_0}{a_0^2}\\ \vdots\\ \dfrac{a_0c_{n-1}-a_{n-1}c_0}{a_0^2}\\ \dfrac{a_0c_n-a_nc_0}{a_0^2}\end{bmatrix}u \quad (2-14)$$

$$y=[1,0,\cdots,0]\boldsymbol{x}+\frac{c_0}{a_0}u \quad (2-15)$$

同样，严格正则的系统，其输入系数矩阵 $\boldsymbol{B}$ 成为：

$$\boldsymbol{B} = \begin{bmatrix} \frac{c_1}{a_0} \\ \vdots \\ \frac{c_n}{a_0} \end{bmatrix}$$

输出方程成为：

$$y = [1, 0, \cdots, 0]\boldsymbol{x}$$

交叉分解法的状态变量图如图 2－6 所示。

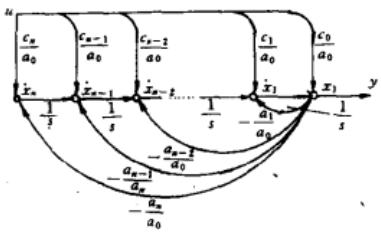


图 2－6　交叉分解法

2.1.3.3　并联分解法（Parallel Resolution）

把（2－8）式的分母多项式分解，设 n 个根为 p_1，p_2，…，p_n。则系统传递函数可写成：

$$G(s) = \frac{Y(s)}{U(s)} = \sum_{i=1}^{n} \frac{K_i}{s - p_i} \qquad (2-16)$$

式中，K_i（$i=1$，2，…，n）是留数。留数的计算可见控制理论的有关书籍。当 p_i 是各不相等的极点时，留数 K_i 由：

$$K_i = \lim_{s \to p_i} G(s) \cdot (s - p_i)$$

计算。这时，若令：

$$\begin{cases} \dot{x}_i = p_i x_i + u \\ y = \sum_{i=1}^{n} K_i x_i \end{cases}$$

就可写成标准形式：

$$\dot{\boldsymbol{x}} = \begin{bmatrix} p_1 & & & 0 \\ & p_2 & & \\ & & \ddots & \\ 0 & & & p_n \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} u \qquad (2-17)$$

$$y = [K_1, K_2, \cdots, K_n]\boldsymbol{x} \qquad (2-18)$$

并联分解法的状态变量图如图 2－7 所示。

当 p_i 有重极点时，如设 p_i 的 l 重根，则留数可由下式计算：

$$K_j = \lim_{s \to p_i} \frac{1}{(j-1)!} \left\{ \frac{d^{j-1}}{ds^{j-1}} (s - p_1)^l \cdot G(s) \right\} \qquad (j = 1, 2, \cdots, l)$$

$$K_i = \lim_{s \to p_1} (s - p_1) G(s) \qquad (i = l+1, \cdots, n)$$

而系数矩阵 A 的对角线元素需用约当矩阵表示。如 p_1 有 l 重根，约当矩阵为 $l \times l$ 阶，即：

$$J_1 = \begin{bmatrix} p_1 & 1 & & 0 \\ & p_1 & 1 & \\ & & \ddots & \ddots \\ 0 & & & p_1 \quad 1 \end{bmatrix}$$

例如某系统有 p_1 重根 4 个，p_2 重根 2 个，p_3 重根 3 个，则系统状态空间表达式为：

$$\dot{x} = \begin{bmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = [K_1, K_2, \cdots, K_9] x$$

式中，J_1、J_2、J_3 为约当矩阵，即：

$$J_1 = \begin{bmatrix} p_1 & 1 & 0 & 0 \\ 0 & p_1 & 1 & 0 \\ 0 & 0 & p_1 & 1 \\ 0 & 0 & 0 & p_1 \end{bmatrix}; J_2 = \begin{bmatrix} p_2 & 1 \\ 0 & p_2 \end{bmatrix}; J_3 = \begin{bmatrix} p_3 & 1 & 0 \\ 0 & p_3 & 1 \\ 0 & 0 & p_3 \end{bmatrix}$$

2.1.3.4 串联分解法

串联分解法又称递推分解法。当系统的零、极点已知时，系统可表示为：

$$G(s) = K \cdot \prod_{i=1}^{n} \frac{s - z_i}{s - p_i} \tag{2-19}$$

串联分解法的状态变量图如图 2-8 所示。可以看到，系统状态空间表达式如下：

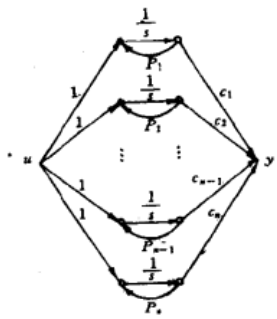


图 2-7 并联分解法

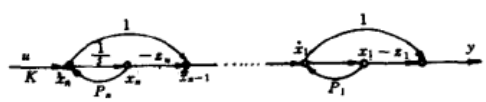


图 2-8 串联分解法

$$\dot{x} = \begin{bmatrix} p_1 & p_2 - z_2 & \cdots & \cdots & p_n - z_n \\ 0 & p_2 & p_3 - z_3 & \cdots & p_n - z_n \\ \vdots & 0 & p_3 & \cdot & \vdots \\ \vdots & \vdots & & & \vdots \\ & & & & p_n - z_n \\ 0 & 0 & \cdots & \cdots & p_n \end{bmatrix} x + \begin{bmatrix} K \\ K \\ \vdots \\ \vdots \\ K \\ K \end{bmatrix} u \tag{2-20}$$

$$y = [p_1 - z_1, p_2 - z_2, \cdots, p_n - z_n] x + Ku \tag{2-21}$$

〔例 2-4〕某系统的传递函数是：

$$G(s) = \frac{2(s+5)}{(s+2)(s+6)(s+10)} \tag{2-22}$$

采用直接分解法，得

$$G(s) = \frac{2s+10}{s^3+18s^2+92s+120}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -120 & -92 & -18 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 10 & 2 & 0 \end{bmatrix} \boldsymbol{x}$$

采用交叉分解法，得：

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -18 & 1 & 0 \\ -92 & 0 & 1 \\ -120 & 0 & 0 \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} 0 \\ 2 \\ 10 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \boldsymbol{x}$$

2.1.4 数学模型的合成

工业生产过程系统常常由许多环节组成，若各个环节的数学模型已经知道，则通过合成的方法可获得整体系统的数学模型。根据组成系统的各环节输入、输出变量之间的连接关系，消去一些中间变量，就能得到整个系统输入变量和输出变量间的关系。

设各环节的数学模型是：

$$\begin{cases} \dot{\boldsymbol{x}}_i = A_i \boldsymbol{x}_i + B_i u_i \\ y_i = C_i \boldsymbol{x}_i + D_i u_i \end{cases} \quad (i = 1, 2, \cdots, n) \tag{2-23}$$

各个环节之间的连接关系用连接矩阵 W 表示。各个环节与整个系统的外部输入（如设定值、扰动等）的关系用连接矩阵 W_0 表示。若组成整个系统的环节数为 n，整个系统外部输入变量数为 p，则 W 是 $n \times n$ 维矩阵，W_0 是 $n \times p$ 维矩阵。整个系统的输入向量 $\boldsymbol{u}$ 可用各环节的输出 y_i 和系统外部输入 Y_0 表示：

$$\boldsymbol{u} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_n \end{bmatrix} = \boldsymbol{W}\boldsymbol{Y} + W_0 Y_0 \tag{2-24}$$

式中，$\boldsymbol{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$

连接矩阵 W 的每个元素 W_{ij} 定义如下：

W_{ij} 表示第 j 个环节对第 i 个环节的连接系数。根据连接关系，W_{ij} 的取值分三类：

①无连接关系：$W_{ij} = 0$。

②直接连接：$W_{ij} = 1$

③通过放大环节连接：若放大环节的放大系数为 k_{ij}，则 $W_{ij} = k_{ij}$。当放大环节是反相器时有 $W_{ij} = -1$。

连接矩阵 W_0 的每个元素 W_{il} 定义为第 l 个外部输入对第 i 个环节的连接系数。取值方

法与 W_{ij} 相类似，不另述。

根据（2－23）式，可列出整个系统的状态空间表达式是：

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t)\\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{2-25}$$

式中，

$$A = \begin{bmatrix} A_1 & & & \\ & A_2 & 0 & \\ & & \ddots & \\ 0 & & & A_n \end{bmatrix}; B = \begin{bmatrix} B_1 & & & \\ & B_2 & 0 & \\ & & \ddots & \\ 0 & & & B_n \end{bmatrix};$$

$$C = \begin{bmatrix} C_1 & & & \\ & C_2 & 0 & \\ & & \ddots & \\ 0 & & & C_n \end{bmatrix}; D = \begin{bmatrix} D_1 & & & \\ & D_2 & 0 & \\ & & \ddots & \\ 0 & & & D_n \end{bmatrix}; x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

用（2－24）代入上式，化简得

$$(I - DW)Y = Cx + DW_0Y_0 \tag{2-26}$$

若矩阵（$I - DW$）是非奇异矩阵，则可求逆矩阵，并得输出方程

$$Y = (I - DW)^{-1}Cx + (I - DW)^{-1}DW_0Y_0 \tag{2-27}$$

及状态方程

$$\dot{x} = [A + BW(I - DW)^{-1}C]x + B[I + W(I - DW)^{-1}D]W_0Y_0 \tag{2-28}$$

而整个系统的输出根据它所连接的信号情况，很方便地由 Y、Y_0 的各元素经组合而成。

〔例 2－5〕 单回路控制系统

若单回路控制系统由控制器、控制阀和对象等三部分组成。如图 2－9 所示。

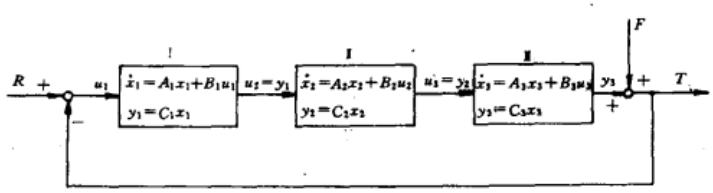


图 2－9　单回路控制系统

各环节状态空间表达式是：

$$\begin{cases}\dot{x}_i = A_ix_i + B_iu_i\\ y_i = C_ix_i\end{cases}$$

列出连接矩阵及外部输入向量：

$$W = \begin{bmatrix}0 & 0 & -1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}; W_0 = \begin{bmatrix}1 & -1\\ 0 & 0\\ 0 & 0\end{bmatrix}; Y_0 = \begin{bmatrix}R\\ F\end{bmatrix}$$

列出整个系统输出 T：

$$T = [0\ 0\ 1]Y + [0\ 1]Y_0$$

由于 $D=0$，（2－28）式成为：

$$\dot{x} = [A + BWC]x + BW_0Y_0 \tag{2-29}$$

对单回路控制系统有：

$$\dot{x}=\begin{bmatrix}A_1 & 0 & -B_1C_3\\ B_2C_1 & A_2 & 0\\ 0 & B_3C_2 & A_3\end{bmatrix}x+\begin{bmatrix}B_1 & -B_1\\ 0 & 0\\ 0 & 0\end{bmatrix}Y_0$$

2.1.5 纯滞后系统的数学模型

若系统含有纯滞后，则系统可以分为不含纯滞后的部分和纯滞后部分。因此系统的传递函数成为：

$$G_p(s)=\frac{Y(s)}{U(s)}=G(s)\cdot e^{-\tau s} \tag{2-30}$$

对含有纯滞后的系统，它的状态空间表达式可用两种途径获得。若把纯滞后部分放在系统输入端，如图2－10（a）所示。其中 $G(s)$ 是不含纯滞后的部分。引入中间变量 $U_{-\tau}(s)$，则有：

$$U_{-\tau}(s)=e^{-\tau s}U(s) \tag{2-31}$$

$$Y(s)=G(s)U_{-\tau}(s) \tag{2-32}$$

$$\because \quad U_{-\tau}(t)=\mathcal{L}^{-1}[U_{-\tau}(s)]=\mathcal{L}^{-1}[e^{-\tau s}U(s)]=u(t-\tau) \qquad t\geqslant\tau \tag{2-33}$$

由于 $G(s)$ 可表示为：

$$\dot{x}(t)=Ax(t)+Bu_{-\tau}(t)$$

$$y(t)=Cx(t)+Du_{-\tau}(t)$$

以（2－33）代入得：

$$\begin{cases}\dot{x}(t)=Ax(t)+Bu(t-\tau) & (2-34)\\ y(t)=Cx(t)+Du(t-\tau) & (2-35)\end{cases}$$

若纯滞后部分放在系统的输出端，如图2－10b所示。同样引入变量 $Y_{+\tau}(s)$，得：

$$Y(s)=e^{-\tau s}Y_{+\tau}(s) \tag{2-36}$$

及

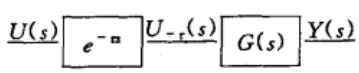
$$Y_{+\tau}(s)=G(s)U(s) \tag{2-37}$$

$$\because \quad y(t)=\mathcal{L}^{-1}[Y(s)]=\mathcal{L}^{-1}[e^{-\tau s}Y_{+\tau}(s)]=y_\tau(t-\tau) \tag{2-38}$$

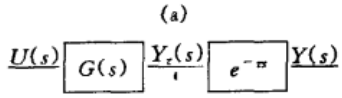
或：

$$y_\tau(t)=y(t+\tau)$$

$\because G(s)$ 可表示为：



(a)



(b)

图2－10　纯滞后的不同位置

$$\dot{x}(t)=Ax(t)+Bu(t) \tag{2-39}$$

$$y_\tau(t)=Cx(t)+Du(t)$$

或：

$$y(t)=Cx(t-\tau)+Du(t-\tau) \tag{2-40}$$

(2－34)(2－35)式与(2－39)(2－40)式都可以用于描述含有纯滞后的系统。对于严格正则系

统，$D=0$，则系统的描述可以简化。当含有纯滞后的各子系统合成时，希望采用(2－39)及(2－40)式。当输入变量作阶跃变化，则可采用(2－34)及(2－35)式。

〔例2－6〕具有纯滞后的闭环系统

如图2－11所示的闭环系统由具有纯滞后分别为 τ_1、τ_2、τ_3 的子系统Ⅰ、Ⅱ、Ⅲ组成。

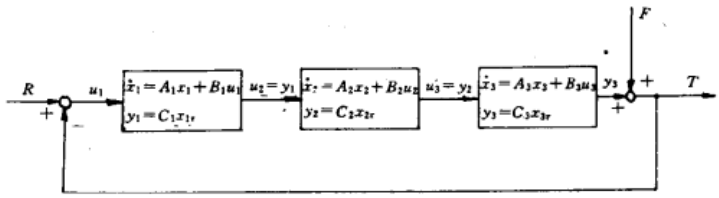


图2－11　纯滞后闭环系统

为简化表示，令 $x_{i\tau}=x_i(t-\tau_i)$。

各子系统的状态空间表达式分别为：

$$\text{Ⅰ}:\begin{cases}\dot{x}_1 = A_1x_1 + B_1u_1\\ y_1 = C_1x_{1\tau}\end{cases};\ \text{Ⅱ}:\begin{cases}\dot{x}_2 = A_2x_2 + B_2u_2\\ y_2 = C_2x_{2\tau}\end{cases};\ \text{Ⅲ}:\begin{cases}\dot{x}_3 = A_3x_3 + B_3u_3\\ y_3 = C_3x_{3\tau}\end{cases}$$

各子系统的输入和输出之间有：

$$u_1 = R-(y_3+F);\ u_2 = y_1;\ u_3 = y_2;\ T = F + y_3 = F + c_3x_{3\tau}$$

化简得：

$$\dot{x} = \begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}A_1 & 0 & 0\\ 0 & A_2 & 0\\ 0 & 0 & A_3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}0 & 0 & -B_1C_3\\ B_2C_1 & 0 & 0\\ 0 & B_3C_2 & 0\end{bmatrix}\begin{bmatrix}x_{1\tau}\\ x_{2\tau}\\ x_{3\tau}\end{bmatrix} + \begin{bmatrix}B_1 & -B_1\\ 0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}R\\ F\end{bmatrix}$$

与例2－5的结果比较可以看到，当纯滞后为零时，有：$x_{1\tau}=x_1$；$x_{2\tau}=x_2$；$x_{3\tau}=x_3$。

可得到与例2－5相同的结果。

对于含有纯滞后的系统数学模型的合成可以用类似于2.1.4的方法，把输出信号均认为是含纯滞后的输出。即 $U=WY_\tau+W_0Y_0$ 及 $Y_\tau=Cx_\tau$。在此处不作详述。

2.2　数值积分算法

在生产实际和科学研究中，所遇到的微分方程往往较复杂，在很多情况下不可能给出解的解析表达式。另一方面，工程应用中，也仅在一定精确度条件下需微分方程的数值解。这种求取微分方程数值解的算法是数值积分法。

2.2.1　数值积分算法的基本思想

以一阶微分方程 $\frac{dy}{dt}=f(t,y)\quad a\leqslant t\leqslant b$ 及初值为 $y(a)=y_a$ 的数值解计算方法为例，说明数值积分算法的基本思想。

(1)离散化

求取在区间〔a,b〕上连续函数 $f(t,y)$ 的数值解的问题，实质上是求取在区间〔a,b〕上若

干离散点 t_0、t_1、t_2、…、t_n($a \leqslant t_0 < t_1 < t_2 < \cdots < t_n \leqslant b$)处解 $y(t)$ 的近似值。离散化则是把区间$[a,b]$离散为 n 个时间段。通常取相等的时间段,即:

$$t_1 - t_0 = t_2 - t_1 = \cdots = t_n - t_{n-1} = h$$

称 h 为积分步长。在数值积分计算过程中,若步长不发生变化,则称为定步长。如果步长随计算进程发生变化,则称为变步长。

离散化使我们把求解连续函数解的问题转化为在给定的 $n+1$ 个点上的近似的差分方程的初值问题。

在点 t_i 的导数$y'(t_i)$ 可以近似地用差商来表示:

$$y'(t_i) \approx \frac{y_{i+1} - y_i}{h}$$

从而把微分方程初值问题转化为差分方程初值问题:

$$\begin{cases} \dfrac{y_{i+1} - y_i}{h} = f(t_i, y_i) & i = 0, 1, \cdots \\ y(a) = y_a \end{cases}$$

式中,y_i 是$y(t)$ 在 t_i 处的近似值。

应当指出,用差商近似导数的方法不是唯一的。一种有效的方法是采用积分的方法。

(2)求取积分值

根据微分方程 $\dfrac{dy}{dt} = f(t,y)$,两边积分得:

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t,y)dt$$

或写出递推形式:

$$y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t,y)dt \qquad (2-41)$$

上式表明,当 $y(t_i)$ 已知时,若积分项可以求得,则通过上式的计算可以求出 $y(t_{i+1})$。当初值 $y(a)$ 已知,求取 $n+1$ 个点处数值解 $y(t_i)(i = 0,1,\cdots n)$ 就转化为求取相应的积分项 $\int_{t_i}^{t_{i+1}} f(t,y)dt$。

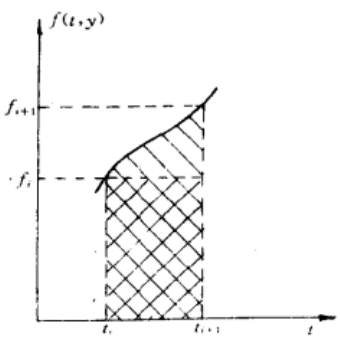


图 2-12 积分项的物理意义

如图 2-12 所示,积分项数值的物理意义就是求取 $f(t,y)$ 在 t_i 到t_{i+1}间阴影区的面积。采用数值积分算法能够求取该面积的近似值。不同的算法所获得的近似值不相同。例如可用矩形面积来近似,也可以用梯形面积来近似。

(3)计算近似解

当初值已知,积分项通过积分算法求得时,微分方程的近似解即可求得:

$$\begin{cases} y(t_{i+1}) = y(t_i) + \int_{t_i}^{t_{i+1}} f(t,y)dt \\ y(a) = y_a \end{cases}$$

用 y_i 表示$y(t)$ 在 t_i 点处的近似值,用 hf_i 表示积分项的近似值,上式改写为:

$$\begin{cases} y_{i+1} \approx y_i + hf_i \\ y(a) = y_a \end{cases} \tag{2-42}$$

我们知道，在一个点 t_i 的附近，$y(t)$ 可以用泰勒级数展开成多项式为：

$$y(t_i + h) \approx y(t_i) + hy'(t_i) + \frac{h^2}{2!}y''(t_i) + \cdots + \frac{h^p}{p!}y^{(p)}(t_i) \tag{2-43}$$

它是 $y(t)$ 的同次数的近似多项式中最好的。若取前二项，即可得(2－42)式。

2.2.2 数值积分算法的一些基本概念

在数值积分算法的介绍前，先说明一些有关的基本概念。

2.2.2.1 单步法和多步法

数值积分算法是一种递推算法。如(2－42)式所示，它由初值 y_a 计算 y_1，由 y_1 计算 y_2，以此类推进行计算。如果算法中，为了计算后一个点 t_{i+1} 的值 y_{i+1}，只需要知道前一点 t_i 的值 y_i，这种算法称为单步法。如上述的(2－42)式的算法就是单步法。若计算 y_{i+1}，需要知道 t_i 及其过去时刻 t_{i-1}、t_{i-2}、…的数据，这种算法称为多步法。线性多步法算法的一般形式是：

$$\begin{aligned} y_{i+1} = {} & a_0 y_i + a_1 y_{i-1} + \cdots + a_k y_{i-k} \\ & + h(b_{-1}f_{i+1} + b_0 f_i + b_1 f_{i-1} + \cdots + b_k f_{i-k}) \end{aligned} \tag{2-44}$$

由于单步法只要根据初值及算法的递推公式进行计算，因此，它是自启动的算法，多步法则因要前几步的数据，因此开始递推计算时，由于缺少数据而无法启动。单步法的另一个优点是后一步步长与前一步步长间没有联系，可以在计算过程中随时根据需要改变步长。例如在计算控制系统的阶跃响应时，过程开始段变化较大，可选用较小的步长，而在进入稳态段时变化平缓，可选用较大的步长。这样做可以既保证计算精度，又不至于增加计算时间。因此，单步法可以采用变步长算法。多步法只能采用定步长。

多步法的优点是计算过的数据在下一步计算时还可以使用，在同样精度条件下，计算工作量明确少于单步法。因此，在大型工程研究中，例如精馏塔、合成塔、裂解炉等计算过程中被广泛采用。

2.2.2.2 显式和隐式算法

在计算 y_{i+1} 时，若所采用的算式右边各项不包含未知量 y_{i+1} 或其导数时，这种算法称显式算法，如(2－42)式及当 $b_{-1}=0$ 时的(2－44)式就是显式算法。若包含未知量 y_{i+1} 或其导数项，则这种算法称隐式算法。如 $b_{-1}\neq 0$ 时(2－44)式就是一例。

采用显式算法，由于等号右边的量均已知，因此可以直接计算。采用隐式算法，则要先假设一个初值。通过预报公式计算第一次迭代值 y_{i+1}，然后，逐步进行校正，使解收敛于精确解。因此计算工作量大，但由于采用迭代算法，因此，可以减小误差，提高计算精度。

2.2.2.3 算法的截断误差和阶数

当方程的解用泰勒公式展开时，有

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \cdots + \frac{h^p}{p!}y^{(p)} + \cdots \tag{2-45}$$

因此，算法的截断误差是指该算法计算所得近似解与方程的精确解之间的误差，它可用泰勒公式的余项表示。如取泰勒公式的前二项，就得到了(2－42)的公式，这时截断误差 ε

$$\varepsilon = o(h^2)$$

取公式的前 $p+1$ 项，则截断误差为 $o(h^{p+1})$。若算法的截断误差为 $o(h^{p+1})$，则称该算法是 p 阶的。p 是算法的阶数，它表示该算法对于所有不超过 p 次的多项式都能精确满足。

2.2.2.4　算法的计算稳定性

在算法的迭代计算过程中，若能收敛到精确解，则该算法是计算稳定的。若不能收敛，则称该算法是计算不稳定的。引起算法计算不稳定的因素主要有两个：

①由于连续的微分方程经数值积分算法离散化为差分方程，从而引起特征根发生变化。这是算法本身造成的。

②由于采用的步长不同，初始迭代值选择不当或者选用的计算机字长不够，造成计算的误差，经不断的积累，从而引起计算结果发散。

解决算法的计算不稳定问题，可采取下列措施：选用足够字长的计算机进行运算；改变步长；更换初始迭代值。

应该指出，对于同一个求解的问题，采用不同的数值积分算法，可能会有不同的稳定条件。对于同一种算法，不同步长计算，也会有可能出现不同的稳定条件。采用同一种算法，相同的步长去求解不同的问题也可能会有不同的稳定条件。

2.2.3　数值积分算法

为了工程应用的需要，掌握几种常用的数值积分算法是必要的。工程计算时，若采用某种数值积分算法，不能得到收敛解，除了可以更改步长、更换初始迭代值外，还可以采用另一种数值积分算法。

2.2.3.1　欧拉法(Euler)

欧拉法是用矩形面积近似积分项的面积。如图 2－13 所示。根据矩形边的不同取法，可分为前差和后差两类。采用前向差分代替导数项，可得欧拉公式是：

$$Y_{i+1} = y_i + hf_i \tag{2-46}$$

式中，$y_i = y(t_i)$；$f_i = f(t_i, y_i)$。

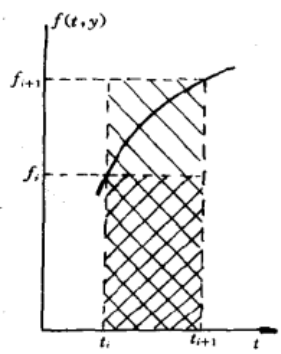


图 2－13　欧拉法

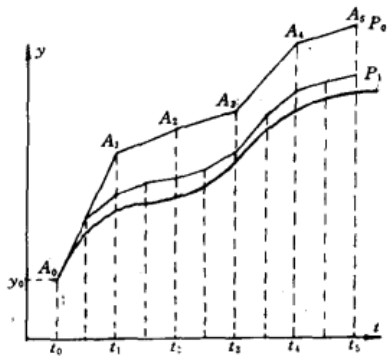


图 2－14　欧拉法计算过程

图 2－14 是欧拉法的求解过程。从 (t_0, y_0) 点出发求 (t_1, y_1) 的过程，实际上是从 (t_0, y_0) 出发，沿方向场在该点的方向 f_0 画一线段，它与直线 $t = t_1$ 的交点，便是所求的 (t_1, y_1)。其余各点依次类推。用 $A_0, A_1, \cdots, A_n$ 分别代表这些点，顺序连接这些点组成一条折线，记为 P_0。若把步长缩小一半，同样从 (t_0, y_0) 出发，可以作出另一条折线 P_1。可以看到折线 P_1 比折线 P_0 更接近于实际曲线。因此，欧拉法是用折线序列来逼近微分方程精确解的。故欧拉法又称折线法。

欧拉法是最简单的数值积分算法。它具有下列特点：

①自启动的单步法。

②运算速度快，方法简单，计算精度低。

③算法是一阶的。截断误差 ε

$$\varepsilon \approx \frac{h^2}{2!} f'_i$$

$$\because \quad f'_i \approx \frac{f_{i+1} - f_i}{h}$$

$$\therefore \quad \varepsilon \approx \frac{h}{2}(f_{i+1} - f_i) \tag{2-47}$$

上式表明欧拉法的截断误差 ε 的大小与步长 h 成正比。减小步长，截断误差减小，近似解逼近精确解。

〔例 2-7〕设 $\frac{dy}{dt} = \lambda y; y(0) = 1$

根据欧拉公式有：

$$y_{i+1} = y_i + hf_i = y_i + h \cdot \lambda y_i = (1 + \lambda h) y_i \quad (i = 0, 1, 2, \cdots)$$

$$\because \quad y(0) = 1$$

$$\therefore \quad y(1) = y(0)(1 + \lambda h) = (1 + \lambda h)$$

$$y(2) = y(1)(1 + \lambda h) = (1 + \lambda h)^2$$

……

其通式：

$$y(i) = (1 + \lambda h)^i$$

在 $t = t_n = nh$ 时

$$y(n) = (1 + \lambda h)^n = [(1 + \lambda h)^{\frac{1}{\lambda h}}]^{\lambda t}$$

根据极限定理，当 $h \to 0$ 即 $n \to \infty$ 时有：$y(n) = y_t$ 趋近于 $e^{\lambda t}$，即

$$\lim_{h \to 0} y(n) = e^{\lambda t}$$

方程的解析解是 $y_t = e^{\lambda t}$

可见，当步长足够小时，欧拉法的计算结果是逼近精确解的。

2.2.3.2 龙格—库塔法(Runge - Kutta)

龙格—库塔算法是基于下列构思：

①在一个点(如 t_i)附近，$y(t)$ 的同次数的近似多项式中，以泰勒多项式为最好。即：

$$y(t_{i+1}) = y(t_i + h) \approx y(t_i) + hy'(t_i) + \cdots + \frac{h^p}{p!} y^{(p)}(t_i) \tag{2-48}$$

②由于 $y(t)$ 的高阶导数不易求取，因此采用间接的方法使用泰勒公式。龙格—库塔算法是用 n 个点上函数值 f 的线性组合来代替 f 的导数，然后，按泰勒公式展开，根据相应项系数相等，确定其线性组合中各项系数，可以看到，由于采用线性组合代替导数项，避免了不易计算导数的困难，同时，因采用系数相等进行计算的，因此可以保证足够的精度。

龙格—库塔算法的基本公式是

$$y_{i+1} = y_i + \sum_{j=1}^{v} w_j k_j \tag{2-49}$$

式中

$$k_j = hf(t_i + c_j h, y_i + \sum_{l=1}^{j-1} a_{jl} k_l) \quad c_1 = 0; j = 1, 2, \cdots, v \tag{2-50}$$

w_j、c_j 及 a_{jl} 均是待定系数。

下面以 $v=2$ 的二阶龙格—库塔公式说明系数是如何被确定的。

当 $v=2$ 时，龙格—库塔公式成为：

$$y_{i+1} = y_i + w_1k_1 + w_2k_2$$
$$k_1 = hf(t_i, y_i) = hf_i$$
$$k_2 = hf(t_i + c_2h, y_i + a_{21}k_1)$$

把 k_2 以 $f(t_i, y_i)$ 的形式展开得：

$$k_2 = h[f(t_i, y_i) + c_2hf'_{ti}(t_i, y_i) + a_{21}k_1f'_{yi}(t_i, y_i)]$$
$$= hf_i + h^2[c_2f'_{ti} + a_{21}f_if'_{yi}]$$

代入，得

$$y_{i+1} = y_i + w_1hf_i + w_2hf_i + w_2h^2[c_2f'_{ti} + a_{21}f_if'_{yi}] \tag{2-50}$$

以泰勒公式展开为：

$$y_{i+1} = y_i + hf_i + \frac{h^2}{2!}f'_i + \cdots$$
$$= y_i + hf_i + \frac{h^2}{2!}[f'_{ti} + f'_{yi}f_i] \tag{2-51}$$

(2－50)与(2－51)的相应项系数相等，得：

$$\begin{cases} w_1 + w_2 = 1 \\ w_2c_2 = \dfrac{1}{2} \\ a_{21}w_2 = \dfrac{1}{2} \end{cases} \tag{2-52}$$

(2－52)式有四个可供调整的参数，方程个数仅三个，因此，可有无数组解满足(2－52)的要求。可先确定一个参数，然后求解其余三个参数。典型的三个二阶龙格—库塔公式如下：

①$c_2=\frac{1}{2}$；$w_1=0$；$w_2=1$；$a_{21}=\frac{1}{2}$时得中点公式：

$$y_{i+1} = y_i + hf(t_i + 0.5h, y_i + 0.5f_i) \tag{2-53}$$

②$c_2=1$；$w_1=\frac{1}{2}$；$w_2=\frac{1}{2}$；$a_{21}=1$ 时得改进欧拉公式：

$$y_{i+1} = y_i + 0.5h[f(t_i, y_i) + f(t_i + h, y_i + hf_i)] \tag{2-54}$$

③$c_2=\frac{1}{3}$；$w_1=\frac{1}{4}$；$w_2=\frac{3}{4}$；$a_{21}=\frac{2}{3}$时得海恩公式(Heun)：

$$y_{i+1} = y_i + 0.25h[f(t_i, y_i) + 3f(t_i + 2/3h, y_i + 2/3hf_i)] \tag{2-55}$$

当 $v=4$ 时，可得四阶龙格—库塔公式。与上述情况相似，也可有各种系数的组合。典型的应用公式是：

$$y_{i+1} = y_i + 1/6[k_1 + 2k_2 + 2k_3 + k_4] \tag{2-56}$$
$$k_1 = hf(t_i, y_i)$$
$$k_2 = hf(t_i + 0.5h, y_i + 0.5k_1)$$
$$k_3 = hf(t_i + 0.5h, y_i + 0.5k_2)$$
$$k_4 = hf(t_i + h, y_i + k_3)$$

小舍入误差的吉尔公式 Gear 是：

$$y_{i+1} = y_i + \frac{1}{6}[k_1 + (2-\sqrt{2})k_2 + (2+\sqrt{2})k_3 + k_4] \tag{2-57}$$

$$k_1 = hf(t_i, y_i)$$

$$k_2 = hf\left(t_i + \frac{h}{2}, y_i + \frac{1}{2}k_1\right)$$

$$k_3 = hf\left[t_i + \frac{h}{2}, y_i + \frac{\sqrt{2}-1}{2}k_1 + \left(1 - \frac{\sqrt{2}}{2}\right)k_2\right]$$

$$k_4 = hf\left[t_i + h, y_i - \frac{\sqrt{2}}{2}k_2 + \left(1 + \frac{\sqrt{2}}{2}\right)k_3\right]$$

〔例2-8〕微分方程 $\frac{dy}{dt} = f(t,y) = yt, y(0) = 1$

其解析解 $y(t) = e^{\frac{t^2}{2}}$

在 $t = 0.1$ 时，精确解是 $y(0.1) = e^{\frac{0.1^2}{2}} \approx 1.005012521$

用二阶龙格—库塔公式，例如用中点公式计算，取步长 $h = 0.1$，得

$$y(0.1) = y(0) + h\left(t_0 + \frac{h}{2}\right)\left(y_0 + \frac{h}{2}f_0\right)$$

$$= 1 + 0.1 \times 0.05 \times 1 = 1.005$$

用四阶龙格—库塔公式，同样取 $h = 0.1$。

$$k_1 = hf(t_0, y_0) = h \cdot y_0 t_0 = 0.1 \times 1 \times 0 = 0$$

$$k_2 = hf\left(t_0 + \frac{h}{2}, y_0 + \frac{1}{2}k_1\right) = 0.1 \times 1 \times 0.05 = 0.005$$

$$k_3 = hf\left(t_0 + \frac{h}{2}, y_0 + \frac{1}{2}k_2\right) = 0.1 \times 0.05 \times 1.0025 = 0.0050125$$

$$k_4 = hf(t_0 + h, y_0 + k_3) = 0.1 \times 0.1 \times 1.0050125 = 0.010050125$$

$$\therefore y(0.1) = y(0) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$= 1 + (0 + 2 \times 0.05 + 2 \times 0.0050125 + 0.010050125)/6 = 1.005012521$$

可见采用龙格—库塔算法的阶数提高，计算精确度提高。用 $N(v)$ 表示阶数 v 的龙格—库塔算法可达到的精确度等级，则有

$$N(1) = 1; N(2) = 2; N(3) = 3; N(4) = 4; N(5) = 4;$$

$$N(6) = 5; N(7) = 6; N(8) = 6; N(9) = 7$$

可以看到，阶数的提高，精确度虽有提高，但计算工作量大大增加，所以，在工程应用中，常用四阶龙格—库塔算法。

上面介绍的龙格—库塔算法都采用显式公式。可以看到，该算法是自启动的单步算法。常常用作同阶数的多步法的启动算法。由于是单步法，因此可根据精确度要求，改变步长，以提高计算效率。

算法的相对截断误差 $\varepsilon\%$ 与步长 h 的 v 次幂成正比：$\varepsilon\% \propto h^v$。如四阶龙格—库塔算法的相对截断误差 $\varepsilon\%$ 与 h^4 成正比。

2.2.3.3 简单的预报校正算法

如图2-11所示，若积分项用梯形面积近似，则

$$\int_{t_i}^{t_{i+1}} f(t,y)dt \approx \frac{h}{2}[f(t_i,y_i) + f(t_{i+1},y_{i+1})]$$

$$\therefore \quad y_{i+1} = y_i + \frac{h}{2}f(t_i,y_i) + \frac{h}{2}f(t_{i+1},y_{i+1})$$

上式是隐式公式，可写成：

预报公式：$y_p = y_i + hf(t_i,y_i)$ (2－58)

校正公式：$y_c = y_i + hf(t_{i+1},y_p)$ (2－59)

计算结果：$y_{i+1} = (y_p + y_c)/2$ (2－60)

该算法具有下列特点：

①采用隐式公式，精确度较高。其截断误差是 $o(h^3)$，算法是二阶的。与欧拉法相比，精度确有提高。一般来说，隐式算法较相应的显式算法精确度要高。

②校正过程可用迭代过程来提高精确度。迭代结束的条件可用 $|y_c - y_p| \leqslant E$，E 是所需的迭代精确度。迭代的方法有两种，框图见图 2－15。

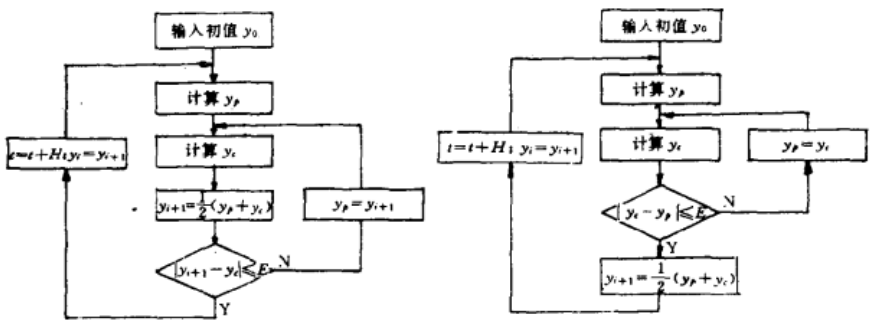


图 2－15　迭代过程框图

③可以证明，该算法是恒稳格式。

2.2.3.4　哈明法(Hamming)

哈明法是为提高算法的计算稳定性而提出的一种多步的预报校正算法。其计算公式是：

$$y_{i+1} = y_{i-3} + 4/3h[2f_i - f_{i-1} + 2f_{i-2}] \quad (2-61)$$

$$y_{i+1} = [9y_i - y_{i-2} + 3h(f_{i+1} + 2f_i - f_{i-1})]/8 \quad (2-62)$$

(2－61)称预报公式，(2－62)称校正公式。上述公式提供的算法是四阶的。由于不能自启动，因此，常用四阶龙格—库塔法启动。本算法的计算步骤如下：

①根据初值，用四阶龙格—库塔法计算前三步的数据。

②用预报公式计算预报值 y_{i+1}，记为 P_{i+1}。

$$P_{i+1} = y_{i-3} + 4/3h[2f_i - f_{i-1} + 2f_{i-2}] \quad (2-63)$$

③用修正公式计算修正项，记为 M_{i+1}。

$$M_{i+1} = P_{i+1} - 112/121(P_i - C_i) \quad (2-64)$$

式中，C_i 是校正公式的校正项，由上一次迭代求出。当 $i=3$ 时，C_3 未求出，用 $C_3 = P_3$ 代入。

④计算校正项 C_{i+1}。

$$C_{i+1} = [9y_i - y_{i-2} + 3h(M'_{i+1} + 2f_i - f_{i-1})]/8 \quad (2-65)$$

式中，$M'_{i+1} = f(t_{i+1}, M_{i+1})$。

⑤计算结果。

$$y_{i+1} = C_{i+1} + 9/121(P_i - C_i) \tag{2-66}$$

$$f_{i+1} = f(t_{i+1}, y_{i+1}) \tag{2-67}$$

本算法也可用迭代计算过程。迭代结束条件是 $|P_{i+1} - C_{i+1}| \leqslant E$。$E$ 是迭代精确度。与2.2.3.3相似，也可有两种迭代结构，此处不另述。

本算法的特点：

①改进了米尔恩算法(Milne)，具有较好的计算稳定性。

②算法的截断误差 ε 与 h^5 成正比。根据算法阶数与算法截断误差中 h 的幂次减1相等的规则，可知，本算法是四阶的。其初始的三步计算应用相同阶的自启动算法，如四阶龙格—库塔法计算。

③采用预报校正计算的工作量较相同阶数的龙格—库塔法要少，因此工程应用中，尤其是计算工作量大的数值积分计算中，如精馏塔、合成塔等操作参数计算中被广泛采用。

2.2.3.5 阿达姆斯法(Adams)

阿达姆斯算法是多步算法。根据(2-44)式，它可分为显式($b_{-1}=0$)及隐式($b_{-1}\neq 0$)两种。常用的四阶阿达姆斯算法如下：

$$y_{i+1} = y_i + h[55f_i - 59f_{i-1} + 37f_{i-2} - 9f_{i-3}]/24 \tag{2-68}$$

$$y_{i+1} = y_i + h[9f_{i+1} + 19f_i - 5f_{i-1} + f_{i-2}]/24 \tag{2-69}$$

(2-68)是显式公式，(2-69)是隐式公式。通常用显式公式进行预报，用隐式公式进行校正。其迭代计算步骤如下：

①用同阶的自启动算法计算前三步数据。

②用(2-68)预报。计算 $y_{i+1}^{(0)}$，迭代次数 k 置0。

③用(2-69)校正。得 $y_{i+1}^{(k+1)}$。

④符合迭代终止条件，转⑥。

⑤ $k+1 \to k$，$f(t_{i+1}, y_{i+1}^{(k)}) \to f_{i+1}^{(k)}$，转③。

⑥ $y_{i+1}^{k} \to y_{i+1}$。

阿达姆斯法适用于有光滑解的非病态方程。它的计算是条件稳定的。

2.2.4 算法的计算稳定性

算法的计算稳定性指采用该算法求解某一微分方程时，计算结果能否收敛到精确解，若能收敛，则称该算法对该方程是计算稳定的。如果收敛应满足某些条件，则称条件稳定。如果对各类方程，算法都是稳定的，则称为绝对稳定或恒稳。

对于一般初值问题，由于微分方程往往较复杂，较难讨论算法的稳定性，所以常用称为"试验"方程的简单方程进行讨论。若某算法对试验方程还是不稳定的，则就很难应用于其他场合。采用试验方程若某算法是恒稳的，不一定对其他方程也是恒稳，但用试验方程讨论，可在一定程度上反映数值积分算法的计算稳定性的程度。试验方程是：

$$\frac{dy}{dt} = \lambda y \qquad \mathrm{Re}\lambda < 0 \tag{2-70}$$

(1)欧拉法

由于数值积分算法是把微分方程的初值问题转化为差分方程，因此，算法的计算稳定性就是讨论差分方程特征根是否在稳定域内的问题。欧拉法的差分方程是：

$$y_{i+1} = y_i + hf_i$$

以试验方程代入得

$$y_{i+1} = y_i + \lambda h y_i$$

经 z 变换得：

$$zY(z) = (1 + \lambda h)Y(z)$$

即特征根为：$z = 1 + \lambda h$

根据特征根的稳定条件 $|z| < 1$，得：

$$|z| = |1 + \lambda h| < 1$$

设原微分方程特征根 $\lambda = \alpha + j\beta$，则有：

$$|1 + \alpha h + j\beta h| < 1$$

或

$$(1 + \alpha h)^2 + \beta^2 h^2 < 1$$

化简得：

$$\left(\alpha + \frac{1}{h}\right)^2 + \beta^2 < \frac{1}{h^2}$$

这表明，在〔z〕平面的稳定域（单位圆）映射到〔λ〕平面也是一个圆。其圆心在 $\left(-\frac{1}{h}, 0\right)$，半径是 $\frac{1}{h}$。当 $h \to 0$ 时，稳定域成为左半平面。原方程特征根在负实轴上时，有 $\lambda = -\alpha$（设 $\alpha > 0$），这时，根据稳定条件可求得：$\frac{2}{h} > \alpha > 0$。由于 $\frac{1}{\alpha}$ 表示原微分方程的时间常数 T，所以，可得步长 h 应满足：$h < 2T$。这是采用欧拉法选择步长的依据。

(2)梯形法

根据梯形法，可得差分方程：

$$y_{i+1} = y_i + \frac{h}{2}[\lambda y_i + \lambda y_{i+1}]$$

经 z 变换得：$zY(z) = Y(z) + \frac{\lambda h}{2}[Y(z) + zY(z)]$

$$z = \frac{1 + \frac{\lambda h}{2}}{1 - \frac{\lambda h}{2}}$$

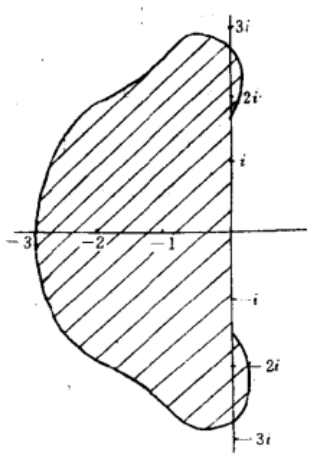


图 2－16　稳定域

根据 $|z| < 1$ 的稳定条件及 $\lambda = \alpha + j\beta$，得：

$$(1 + \frac{\alpha h}{2})^2 < (1 - \frac{\alpha h}{2})^2 \text{ 或 } \alpha h < 0 \text{ ; } \alpha < 0$$

上式表明，原方程的特征根有负实部（$\alpha < 0$）时，采用梯形算法是恒稳的。

(3)四阶龙格－库塔法

对 $\frac{dy}{dt} = \lambda y$，同样可以得出，

$$y_{i+1} = (1 + \lambda y + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24}) y_i$$

即绝对稳定性条件是

$$\left|1 + \lambda h + \frac{(\lambda h)^2}{2} + \frac{(\lambda h)^3}{6} + \frac{(\lambda h)^4}{24}\right| \leqslant 1$$

图 2－16 内阴影部分的区域即是稳定域。

(4)哈明法

以$\frac{dy}{dt}=\lambda y$ 代入(2－62)，得到：

$$(\frac{3}{8}\lambda h-1)y_{i+1}+(\frac{9}{8}+\frac{3}{4}\lambda h)y_i-\frac{3}{8}\lambda h y_{i-1}-\frac{1}{8}y_{i-2}=0$$

特征方程是：

$$(\frac{3}{8}\lambda h-1)\mu^3+(\frac{9}{8}+\frac{3}{4}\lambda h)\mu^2-\frac{3}{8}\lambda h\mu-\frac{1}{8}=0$$

当 $h\to 0$ 时，特征根是：$\mu_1=1$；$\mu_{2,3}=\frac{1\pm\sqrt{33}}{16}$。由于 $|\mu_{2,3}|<1$，这表明哈明法是强稳定的。已经分析得出，当 λ 有负实部时，$h<2.6T$，采用预报校正方法时，$h<0.85T$。

应该指出，在相容性条件成立时，算法的收敛性和稳定性是等价的。当发现采用某种算法运算时有不稳定情况发生时，可以采用下列措施：

①采用另一种数值积分算法。

②减小步长。

③采用在计算过程中能估计误差大小，自动改变步长的变步长算法。

数值积分算法的选择十分重要，一般可按下列原则考虑：

①精度：它受截断误差、舍入误差、累积误差影响。减小步长，截断误差减小，累积误差加大(在一定积分时间内)。采用的算法阶数高，截断误差就小。所用计算机字长越长，舍 λ 误差越小。所以，从所需的总误差出发，应选择合适步长。

②计算速度：对算法每前进一步，若计算次数多，则计算速度就下降。在相同积分时间内，耗机时就长。在一定精度下，选用较大的步长，有利于提高计算速度。

③计算稳定性：不同的数值积分算法有不同的稳定条件，如用四阶龙格一库塔法，步长应满足：$h<2.78T$。

一般应用研究中，常用的数值积分算法是四阶龙格一库塔法、哈明法等。

2.3 数字仿真程序

2.3.1 数字仿真程序的基本结构

数字仿真程序的基本结构由四部分组成：初始化、输入、运算、输出程序。

初始化程序　预置所需数组的维数，对有关单元的复位或清零等。

输入程序　人机对话部分，用于预置初始值、计算步长、计算终止时间(或计算步数)、打印或制图方式、可调参数初值、变量的极限值以及其他需人机对话输入的数据。

运算程序　仿真程序的核心部分。对于连续系统应计算出状态变量、输出变量等数值。对于离散事件系统，除计算状态变量外，还要计算有关的统计特性等。

由于仿真程序的通用性，这类程序应能适用于不同的模型。通过选择，还能选用不同的仿真算法，并应具有多次仿真控制功能。

输出程序　用于把运算结果输出，可用打印机或绘图仪给出相应的曲线数据或图表。对离散事件系统则应给出有关统计特性的图表。

仿真程序通常有下列特点：

①通用性强，能适用于不同的仿真问题。

②可移植性好。采用标准的计算机语言，可适用于不同机型的计算机。

③功能强。具有可以修改参数，如调整步长、更改算法等功能。具有多次仿真的功能。

2.3.2 数值积分算法子程序

本书提供的仿真程序都是用 BASIC 语言编写的，在 IBM—PC 机上运行通过。数值积分算法子程序都是针对 n 阶的状态空间表达式的。当 $n=1$ 则为一阶微分方程。下面介绍的数值积分子程序从 1000 行号开始。

2.3.2.1 欧拉法子程序

根据欧拉法计算公式(2－46)，编制了下列欧拉法子程序。

```
1000  REM EULER INTEGRATE SUBROUTINE
1010  GOSUB 2000
1020  FOR I=1 TO N
1030  Y(I)=Y(I)+H*F(I)
1040  NEXT I
1050  RETURN
```

2000 行号开始的子程序是微分方程组子程序。该程序可以由用户按微分方程组编写，也可由用户输入方程组的系数。例如，欲求解微分方程：$\frac{dy}{dt}=1-y$ 可编写下列仿真模型子程序：

```
2000  F=1-Y
2010  RETURN
```

若用户输入系数值，可见输入程序。

2.3.2.2 龙格—库塔法子程序

为了简化程序，这里给出了有关书籍介绍的龙格—库塔法子程序。它含有语句少，可用于微分方程组的求解。程序清单如下：

```
1000  REM RUNGE-KUTTA INTEGRATE SUBROUTINE
1010  U(1)=.5*H
1020  U(2)=.5*H
1030  U(5)=.5*H
1040  U(3)=H
1050  U(4)=H
1060  S=T
1070  FOR I=1 TO N
1080  P(I)=Y(I)
1090  W(I)=Y(I)
1100  NEXT I
1110  FOR J=1 TO 4
1120  GOSUB 2000
1130  T=S+U(J)
1140  FOR Q=1 TO N
```

```
1150  Y(Q) = U(J) * F(Q) + W(Q)
1160  P(Q) = U(J + 1) * F(Q)/3 + P(Q)
1170  NEXT Q
1180  NEXT J
1190  FOR I = 1 TO N
1200  Y(I) = P(I)
1210  NEXT I
1220  RETURN
```

程序中,T 是时间。2000 行号开始的是仿真模型子程序。

2.3.2.3 哈明法子程序

哈明法是多步法。需用龙格一库塔法启动计算前三步的值。程序清单如下:

```
1000  REM HAMMING INTEGRATE SUBROUTINE
1002  IF M > = 4 THEN 3500
1004  M = M + 1
1006  IF M = 1 THEN 3000
1010  U(1) = .5 * H
1020  U(2) = .5 * H
1030  U(5) = .5 * H
1040  U(3) = H
1050  U(4) = H
1060  S = T
1070  FOR I = 1 TO N
1080  P(I) = Y(I)
1090  W(I) = Y(I)
1100  NEXT I
1110  FOR J = 1 TO 4
1120  GOSUB 2000
1130  T = S + U(J)
1140  FOR Q = 1 TO N
1150  Y(Q) = U(J) * F(Q) + W(Q)
1160  P(Q) = U(J + 1) * F(Q)/3 + P(Q)
1170  NEXT Q
1175  IF J = 4 THEN 3000
1180  NEXT J
1190  FOR I = 1 TO N
1200  Y(I) = P(I)
1210  NEXT I
1220  RETURN
3000  FOR I = 1 TO N
3010  Y(I) = P(I)
```

```
3020  NEXT I
3030  GOSUB 2000
3040  FOR K=1 TO N
3050  E(1,K)=E(2,K)
3060  E(2,K)=E(3,K)
3070  E(3,K)=E(4,K)
3080  E(4,K)=P(K)
3090  E(5,K)=E(6,K)
3100  E(6,K)=E(7,K)
3110  E(7,K)=F(K)
3120  NEXT K
3130  RETURN
3500  FOR K=1 TO N
3510  G(K)=E(1,K)+8/3*H*(E(7,K)-E(6,K)/2+E(5,K))
3520  R(K)=G(K)-112/121*E(8,K)
3530  NEXT K
3540  FOR I=1 TO N
3550  Y(I)=R(I)
3560  NEXT I
3570  GOSUB 2000
3580  FOR K=1 TO N
3590  C=(9*E(4,K)-E(2,K)+3*H*(2*E(7,K)-E(6,K)))/8
3600  S=C+3*H/8*F(K)
3610  E(8,K)=G(K)-S
3620  P(K)=S+9/121*E(8,K)
3630  NEXT K
3640  T=T+H
3650  GOTO 3000
```

其中 1010—1220 是龙格—库塔法子程序。1002—1006 用于确定是否是前三步。3000—3130 是移位。3500—3650 用于哈明法计算。加入 1175 行用于龙格—库塔法计算三步后的数据移位。

程序中符号与哈明法公式中符号对应如下：

G(K)：P_{i+1}；	S：C_{i+1}；	F(K)：M'_{i+1}；
R(K)：M_{i+1}；	E(1,K)：y_{i-3}；	E(2,K)：y_{i-2}；
E(3,K)：y_{i-1}；	E(4,K)：y_i；	E(5,K)：y'_{i-2}；
E(6,K)：y'_{i-1}；	E(7,K)：y'_i；	E(8,K)：$P_{i+1}-C_{i+1}$
P(K)：y_{i+1}		

需要采用迭代计算时，可加入下列程序段：

```
3526  IF ABS(E(8,K))<.001 THEN 3532
```

```
3528  F(K)=SS
3530  GOTO 3522
```

2.3.3 输入子程序段

输入子程序段是数字仿真程序的一部分，不作为子程序。一般可分为两类。当用户编写微分方程组子程序时，由于不必送入状态方程的系数，程序较简单，如下述：

```
10  REM INPUT DATA
15  CLS
20  INPUT"The Number of Equations";N
25  INPUT"The Integrate Step Length";H
30  INPUT"The End Time";TE
35  DIM Y(N),P(N),W(N),F(N)
40  FOR I=1 TO N
45  PRINT "The Start Value of y(";I;")is";
50  INPUT Y(I)
55  NEXT I
60  T=0
65  PRINT"TIME";
70  FOR I=1 TO N
75  PRINT"Y(";I;")";
80  NEXT I
85  PRINT
```

本程序段适用于龙格—库塔法。如为哈明法，35 句改为：

```
35 DIM Y(N),P(N),W(N),F(N),G(N),R(N),E(8,N)
```

数据全部采用 INPUT 语句输入，也可改为其他输入语句，如 LET、READ 等。

程序中符号的意义如下，

N：微分方程个数。

H：积分步长。

TE：积分终止时间。

Y(I)：初值。

积分终止时间 TE 也可用需积分的步数 SN 表示，则有：TE = SN * H。

当输入数据还包括状态空间表达式的各矩阵元素时，可用下列程序段。

```
10  REM INPUT DATA (INCL.MATRIX)
15  CLS
20  INPUT"The Number of State Variables";N
25  INPUT "The Number of Input Variables";NR
30  INPUT"The Number of Output Variables";NP
35  INPUT "The Integrate Step Length";H
40  INPUT "The End Time";TE
45  DIM A(N,N),B(N,NR),C(NP,N),D(NP,NR),Y(N),U(NR),OT(NP),P(N),
W(N),F(N)
```

```
50  FOR I=1 TO N
51  READ Y(I):REM START VALUE OF STATE VARIABLES
52  NEXT I
53  FOR I=1 TO NR
54  READ U(I):REM INPUT VARIABLE VALUE
55  NEXT I
62  FOR I=1 TO N
64  FOR J=1 TO N
66  READ A(I,J):REM STATE MATRIX ELEMENTS
68  NEXT J
70  NEXT I
72  FOR I=1 TO N
74  FOR J=1 TO NR
76  READ B(I,J):REM INPUT MATRIX ELEMENTS
78  NEXT J
80  NEXT I
82  IF NP=0 THEN 104
84  FOR I=1 TO NP
86  FOR J=1 TO N
88  READ C(I,J):REM OUTPUT MATRIX 1 ELEMENTS
90  NEXT J
92  NEXT I
94  FOR I=1 TO NP
96  FOR J=1 TO NR
98  READ D(I,J):REM OUTPUT MATRIX 2 ELEMENTS
100  NEXT J
102  NEXT I
104  T=0
```

除了上述的符号外，程序中其他符号的意义如下：

NR：输入变量个数。

NP：输出变量个数。

U(I)：输入变量的值，当输入变量是时变时，53－55 行删除，并在微分方程组子程序中加入相应语句。

A(I,J)、B(I,J)、C(I,J)、D(I,J)：状态空间表达式中的相应矩阵 A、B、C、D 的元素。

当本程序段用于哈明法时，在 45 句后面增加，“G(N),R(N),E(8,N)”。与前面所述的 35 句更改时增加的内容相同。

输入的数据，部分采用 INPUT 语句，部分则用 READ 语句。这是因为对同一个问题，常常要改变步长、积分终止时间等，而微分方程组中的系数则一般不作更改。

输入部分的程序段除了上述基本内容外，还可以包括作图语句，如在 IBM－PC 机上可以用 SCREEN、LINE、PSET、COLOR 等语句，可参考有关书籍。本章最后给出的仿真程序给出

了有网格点的图形输出方面的程序，为此，在输入部分加入了有关程序。

2.3.4 微分方程组子程序

这里仅介绍通过输入状态空间表达式的有关矩阵元素时，微分方程组子程序的编写。子程序清单如下：

```
2000  REM DEV.EQUATIONS 2
2010  FOR I=1 TO N
2020  FOR J=1 TO N
2030  F(I)=A(I,J)*Y(J)
2040  NEXT J
2050  FOR J=1 TO NR
2060  F(I)=F(I)+B(I,J)*U(J)
2070  NEXT J
2080  NEXT I
2090  RETURN
```

程序中符号如前述，其中，F(I)即微分方程组的导数项，或状态方程中的 $\dot{x}(t)$。Y(J)是微分方程中的变量，或状态方程中的 $x(t)$。

微分方程组子程序在大多数应用中是重新编写的。例如对于双线性、非线性系统就必须重新编写。有些场合，为了研究某些参数(如对象时间常数)的影响，也必须把该参数作为变量列写在微分方程组内，而变量的值则由输入子程序段给出。下面以实例说明编写的方法。

[例 2－9] 为研究某地区在病毒传染情况下，可能受传染人数 x_1，已被传染得病人数 x_2 及已经治愈人数 x_3 变化的情况，可列出下列状态方程：

$$\begin{cases} \dot{x}_1 = -\alpha x_1 x_2 \\ \dot{x}_2 = \alpha x_1 x_2 - \beta x_2 \\ \dot{x}_3 = \beta x_2 \end{cases}$$

式中 α——单位时间内 x_1 的传染系数；

β——单位时间内 x_2 被治愈的比例系数。

据上述方程，可列写出微分方程组子程序：

```
2000  REM DEV.EQUATIONS
2010  F(1)=-ALF*Y(1)*Y(2)
2020  F(3)=BETA*Y(2)
2030  F(2)=-F(1)-F(3)
2040  RETURN
```

[例 2－10] 为研究具有纯滞后的对象在比例积分调节作用下的响应特性，首先列出控制器 $G_c(s)=K_C(1+\frac{1}{T_I s}$ 的状态方程和输出方程：

$$\dot{x}_1 = u_1; y_1 = \frac{K_c}{T_\tau} x_1 + K_c u_1$$

其次列出对象

$$G_p(s) = \frac{Ke^{-\tau s}}{Ts+1}$$

的状态方程和输出方程：

$$\dot{x}_2 = -\frac{1}{T}x_2 + u_2; y_2 = \frac{K}{T}x_{2\tau}$$

式中，$x_{2\tau} = x_2(t - \tau_2)$

根据连接图 2－17，列出连接关系：

$$u_1 = R - F - y_2; u_2 = y_1$$

因此有：

$$\dot{x}_1 = -\frac{K}{T}x_{2\tau} + R - F$$

$$\dot{x}_2 = \frac{K_c}{T_I}x_1 - \frac{1}{T}x_2 - \frac{KK_c}{T}x_{2\tau} + K_cR - K_cF$$

$$y_2 = F + \frac{K}{T}x_{2\tau}$$

列成微分方程组子程序：

```
2000  REM DEV. EQUATION 2
2010  F(1) = -K/T * YT(2,0) + R - F
2020  F(2) = KC/TI * Y(1) - 1/T * Y(2) - K * KC/T * YT(2,0) + KC * R - KC * F
2030  RETURN
```

程序中的 YT(2,0)是 Y(2)移位后得到的值，详见下述的纯滞后子程序。F(1)，F(2)对应于 $\dot{x}_1$ 与 $\dot{x}_2$。Y(1)、Y(2)对应于 x_1、x_2。

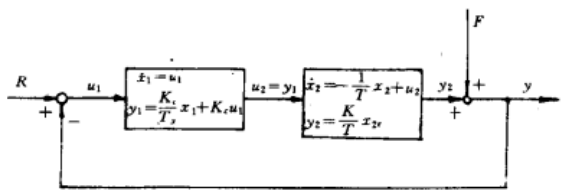


图 2－17 系统连接图

2.3.5 主程序及输出子程序段

与输入子程序段相对应，主程序及输出子程序段也有两种编写方法，下面列出了采用状态矩阵元素时的有关程序。

```
500  REM MAIN AND OUTPUT PROGRAMS 2
505  GOSUB 1000
510  FOR LL=1 TO NP
512  OT(LL)=0
515  FOR DD=1 TO N
520  OT(LL)=OT(LL)+C(LL,DD)*Y(DD)
525  NEXT DD
530  FOR DD=1 TO NR
535  OT(LL)=OT(LL)+D(LL,DD)*UU(DD)
540  NEXT DD
545  NEXT LL
```

```
590  IF T<TE THEN 500
595  END
```

程序中,OT(LL)是输出变量。Y(DD)、UU(DD)是状态变量和输入变量。

输出的数据可以直接打印出来,也可以用响应曲线绘制。下面列出了用响应曲线绘制时的子程序。

```
600  REM DRAW PROGRAMS 2
615  FOR LL=1 TO NP
620  PSET (Z+80,BS-CS*OT(LL))
635  NEXT LL
637  IF NB<>BN THEN 680
640  LOCATE 17+PN,1
645  PRINT INT (T*1000)/1000
650  FOR LL=1 TO NP
652  LOCATE 17+PN,8+(LL-1)*8
653  PRINT INT(OT(LL)*100)/100
654  LOCATE 18+PN,1+(LL-1)*8
655  IF PN=4 THEN LOCATE 17,2+(LL-1)*8
656  PRINT" "
665  NEXT LL
670  PN=PN+1
675  NB=0
680  NB=NB+1
685  Z=Z+1
690  IF PN>4 THEN PN=0
695  RETURN
```

程序中,除了把计算的结果用响应曲线绘制外,还在屏幕的下部显示相应的数据值,程序中的BS、CS是与直角坐标的点数、刻度的大小等有关的数据,由绘制曲线的初始化子程序段给出。BN是打印间隔点数。NB是点数计算。

2.3.6 纯滞后子程序

当系统中含有纯滞后时,需要增加三部分内容。输入部分增加系数矩阵 A_τ、C_τ 的元素输入子程序。输出部分增加计算 $C_\tau x_\tau$ 的部分子程序。运算部分增加计算 $A_\tau x_\tau$ 的微分方程组内容以及纯滞后移位子程序。

假设纯滞后时间为 τ,积分步长为 H,则在数字仿真时应先定义一个数组,用于存放当前时刻以前的数据。数组的大小由 INT(TOR/H+0.5)决定。INT(·)表示对括号内的数取整。TOR即纯滞后 τ。加0.5是用于四舍五入计算。为了便于取出滞后 τ 时刻的数据。数组大小多用一个单元。在有些程序应用时,也可少开一个单元的位置。

纯滞后移位子程序采用先平移,再存取的方法。也可用先取再移再存,这时单元数可少一个。平移指后一内存单元号内的数据移到前一个内存单元中。移动从最小单元号开始,即从单元号1移到单元号0。以此类推,直到最后一个单元号的内容移出。然后,把当前采得的数据放入最后一个单元号中。把0号单元中的内容作为纯滞后 τ 时刻的值取出。在子程序中取

出0号单元的数据是放在微分方程组子程序中进行的。纯滞后移位子程序如下：

```
1500  REM DEAD TIME SHIFT
1505  FOR LL=1 TO N
1510  FOR JJ=0 TO TOR(LL)/H-1
1515  YT(LL,JJ)=YT(LL,JJ+1)
1520  NEXT JJ
1525  YT(LL,TOR(LL)/H)=Y(LL)
1530  NEXT LL
1540  RETURN
```

程序中，Y(LL)是状态变量 x，YT(LL, JJ)是在所开的内存单元中的数据。LL 是状态变量号，从1到N。JJ 是内存单元号。

计算 $A_r x_r$ 的程序段如下：

```
1600  FOR II=1 TO N
1610  FOR JJ=1 TO N
1620  F(II)=F(II)+AT(II,JJ)*YT(II,0)
1630  NEXT JJ
1640  NEXT II
1650  RETURN
```

程序中的 YT(II,0)即是状态变量 x_{II} 的纯滞后以后的数据。

输出部分增加计算 $C_r X_r$ 的子程序段与上述相似，不另述。输入部分增加输入 A_r、C_r 也与输入 A、B、C、D 相似。故也不另述。纯滞后 TOR 的输入程序段如下：

```
212  M=0
214  FOR LL=1 TO N
216  READ TOR(LL):REM DEAD TIME
218  IF M<TOR(LL) THEN M=TOR(LL)
220  NEXT LL
```

由于状态变量数为 N 个，为便于对这 N 个状态变量均设置纯滞后的单元，选用 M 作为最大的纯滞后数值。并根据 M 值决定内存单元的维数。实际应用时，也可以把纯滞后所用的内存单元维数开得足够大，以便于编程。

2.3.7 数字仿真程序举例

为了使读者对数字仿真程序有一个较完整的了解，我们用在 IBM－PC 微机上运行并通过的数字仿真程序为例说明。

本程序可用于对多输入多输出的线性含纯滞后的系统进行数字仿真。输入部分包括键盘输入的数据及用 DATA 语句读入的数据。键盘输入的数据依次有：状态变量数目 N；输入变量数目 NR；输出变量数目 NP；积分步长 H；积分终止时间 TE；打印数据时数据点的间隔 BN；此外，当图形显示时，还要输入刻度的上、下限：RMAX、RMIN。程序提供用户三种输出的显示方式。0：表示用 CRT 显示，1：表示绘制响应曲线，2：表示打印机打印数据。用 DATA 语句读入的数据按下列顺序送入，状态变量初值；输入变量初值；系数矩阵 A、B、C、D 的各元素；纯滞后数值；系数矩阵 A_r、C_r 的各元素。

程序框图如图2－18所示。程序清单见后。程序中所用到的部分变量说明如下：

TR：纯滞后存在标志。TR＝1 表示系统存在纯滞后。

PR：输出方式选择号。PR＝0、1、2 分别为 CRT、图形及数据输出。

PN：行计数，用于数据滚动显示。

数字仿真程序清单：

```
5  REM SIMULATION PROGRAM Ⅲ, COPYRIGHT BY ECUCT, 1985, 1986
10  REM INPUT DATA 2
12  CLS
14  INPUT"The Number of State Variables";N
16  INPUT"The Number of Input Variables";NR
18  INPUT"The Number of Output Variables";NP
20  INPUT"The Integrate Step Length";H
22  INPUT"The Integrate End Time";TE
24  INPUT"Dead Time Exist(Y or N)";A$
26  INPUT"Interval of Printing Points";BN
27  PRINT"0:CRT     1:DRAW     2:PRINT"
28  INPUT"Select Output Mode(0, 1, or 2)";PR
30  A$ = RIGHT$ (A$, 1)
32  IF A$ ="Y"OR A$ ="Y"THEN TR=1
34  IF PR<0 OR PR>2 THEN 27
36  DIM A(N, N), B(N, NR), C(NP, N), D(NP, NR), Y(N), UU(NR), OT(NP)
38  DIM P(N), W(N), F(N), AT(N, N), CT(NP, N), YT(N, 1000), TOR(N)
40  FOR I=1 TO N
42  READ Y(I):REM STATE START VALUES
44  NEXT I
46  FOR I=1 TO NR
48  READ UU(I):REM INPUT VARIABLE VALUES
50  NEXT I
52  FOR I=1 TO N
54  FOR J=1 TO N
56  READ A(I, J):REM STATE MATRIX ELEMENTS
58  NEXT J
60  NEXT I
62  FOR I=1 TO N
64  FOR J=1 TO NR
66  READ B(I, J):REM INPUT MATRIX ELEMENTS
68  NEXT J
70  NEXT I
72  FOR I=1 TO NP
74  FOR J=1 TO N
76  READ C(I, J):REM OUTPUT MATRIX ELEMENTS 1
```

```
 78  NEXT J
 80  NEXT I
 82  FOR I=1 TO NP
 84  FOR J=1 TO NR
 86  READ D(I,J):REM OUTPUT MATRIX ELEMENTS 2
 88  NEXT J
 90  NEXT I
 92  T=0
 94  Z=1
100  IF TR=1 THEN GOSUB 500
102  IF PR=1 THEN GOSUB 600
104  CLS
106  IF PR=1 THEN GOSUB 700
108  NB=BN
200  REM MAIN AND OUTPUT PROGRAMS 2
202  IF PR=1 THEN TE=H*400
206  GOSUB 900
210  IF PR<>1 THEN GOSUB 1800
212  IF PR=1 THEN GOSUB 1500
214  GOSUB 1000
218  GOSUB 900
222  IF PR=1 THEN GOSUB 1500
224  IF PR<>1 THEN GOSUB 1822
226  IF T<TE THEN 214
228  IF PR=1 THEN LPRINT CHR$(27)"W"
230  END
500  REM DEAD TIME INPUT DATA
504  M=0
506  FOR I=1 TO N
508  READ TOR(I):REM DEAD TIME FOR EVERY BLOCK
510  IF M<TOR(I) THEN M=TOR(I)
512  NEXT I
514  M=M/H
518  FOR I=1 TO N
520  FOR J=1 TO N
522  READ AT(I,J):REM DEAD TIME MATRIX ELEMENTS 1
524  NEXT J
526  NEXT I
528  FOR I=1 TO NP
530  FOR J=1 TO N
```

```
532  READ CT(I,J):REM DEAD TIME MATRIX ELEMENTS 2
534  NEXT J
536  NEXT I
538  FOR I=1 TO N
540  FOR J=0 TO M
542  YT(I,J)=0:REM DEAD TIME SHIFT ELEMENTS SET ZERO
544  NEXT J
546  NEXT I
548  RETURN
600  REM OUTPUT FOR DRAW 1
604  INPUT"Range Max of Output figure";RMAX
606  INPUT"Range Min of Output figure";RMIN
608  CS=100/(RMAX-RMIN)
610  NB=BN
612  RETURN
700  REM DRAW INIT FIGURE
702  SCREEN 2,0
704  LINE(80,10)-(80,110)
706  LINE-(480,110)
708  FOR I=-1 TO 9
710  PSET(81,I*10+20)
712  NEXT I
714  FOR I=0 TO 20
716  PSET(I*20+80,109)
718  NEXT I
720  FOR I=1 TO 20
722  FOR J=-1 TO 9
724  PSET(I*20+80,J*10+20)
726  NEXT J
728  NEXT I
730  LOCATE 16,2
732  PRINT"Time"
734  FOR I=1 TO NP
736  LOCATE 16,9+(I-1)*60/NP
738  PRINT"Y(";I;")"
740  NEXT I
742  LOCATE 2,1
744  PRINT SPC(8-LEN(STR$(RMAX)));RMAX
746  LOCATE 14,1
748  PRINT SPC(8-LEN(STR$(RMIN)));RMIN
```

```
750  LOCATE 15, 52
752  RH = INT(H * 400 * 100)/100
754  PRINT SPC(10 - LEN(STR$ (RH)));RH
756  RETURN
800  REM PRINT PROGRAMS 2
802  LOCATE 1, 1
804  PRINT"Time"
806  FOR I = 1 TO NP
808  LOCATE 1, 60/NP * (I - 1) + 10
810  PRINT"Y(";I;")"
812  NEXT I
814  PN = 0
816  IF BN< >NB THEN 858
820  LOCATE 3 + PN, 1
822  DT = INT(T * 1000 + .5)/1000
826  PRINT DT
828  LOCATE 4 + PN, 1
830  IF PN = 19 THEN LOCATE 3, 1
832  PRINT"                "
834  FOR LL = 1 TO NP
836  LOCATE 3 + PN, 8 + (LL - 1) * (60/NP)
838  OTT = INT(OT(LL) * 1000 + .5)/1000
842  PRINT OTT
844  LOCATE 4 + PN, 9 + (LL - 1) * (60/NP)
846  IF PN = 19 THEN LOCATE 3, 9 + (LL - 1) * (60/NP)
848  PRINT"                "
850  IF PN>19 THEN PN = 0
852  NEXT LL
854  PN = PN + 1
856  NB = 0
858  NB = NB + 1
860  IF PN>19 THEN PN = 0
862  RETURN
900  REM OUTPUT CALCULATE PROGRAMS
902  FOR LL = 1 TO NP
904  OT(LL) = 0
906  FOR DD = 1 TO N
908  OT(LL) = OT(LL) + C(LL, DD) * Y(DD)
910  NEXT DD
912  FOR DD = 1 TO NR
```

```
914  OT(LL) = OT(LL) + D(LL, DD) * UU(DD)
916  NEXT DD
918  NEXT LL
920  IF TR = 1 THEN GOSUB 1650
922  IF TR = 1 THEN GOSUB 950
924  RETURN
950  REM DEAD TIME OUTPUT
952  FOR LL = 1 TO NP
954  FOR DD = 1 TO N
956  OT(LL) = OT(LL) + CT(LL, DD) * YT(DD, 0)
958  NEXT DD
960  NEXT LL
962  RETURN
1000  REM RUNGE - KUTTA METHODS
1010  U(1) = .5 * H
1020  U(2) = .5 * H
1030  U(5) = .5 * H
1040  U(3) = H
1050  U(4) = H
1060  S = T
1070  FOR I = 1 TO N
1080  P(I) = Y(I)
1090  W(I) = Y(I)
1100  NEXT I
1110  FOR J = 1 TO 4
1120  GOSUB 2000
1130  T = S + U(J)
1140  FOR Q = 1 TO N
1150  Y(Q) = U(J) * F(Q) + W(Q)
1160  P(Q) = U(J + 1) * F(Q)/3 + P(Q)
1170  NEXT Q
1180  NEXT J
1190  FOR I = 1 TO N
1200  Y(I) = P(I)
1210  NEXT I
1220  RETURN
1500  REM DRAW PROGRAM
1502  FOR LL = 1 TO NP
1504  PSET(Z + 80, 110 - CS * (OT(LL) - RMIN))
1506  NEXT LL
```

```
1508  Z=Z+1
1510  IF NB<>BN THEN 1540
1512  LOCATE 17+PN,1
1514  PRINT INT(T*1000)/1000
1516  LOCATE 18+PN,1
1518  IF PN=4 THEN LOCATE 17,1
1520  PRINT"              "
1522  FOR LL=1 TO NP
1524  LOCATE 17+PN,8+(LL-1)*(60/NP)
1526  PRINT INT(OT(LL)*1000)/1000
1528  LOCATE 18+PN,8+(LL-1)*(60/NP)
1530  IF PN=4 THEN LOCATE 17,8+(LL+1)*(60/NP)
1532  RPINT"              "
1534  NEXT LL
1536  PN=PN+1
1538  NB=0
1540  NB=NB+1
1542  IF PN>4 THEN PN=0
1544  RETURN
1600  REM DEAD TIME AT CALCULATE
1602  FOR II=1 TO N
1604  FOR JJ=1 TO N
1606  F(II)=F(II)+AT(II,JJ)*YT(JJ,0)
1608  NEXT JJ
1610  NEXT II
1612  RETURN
1650  REM DEAD TIME YT SHIFT
1652  FOR LL=1 TO N
1654  FOR DD=0 TO TOR(LL)/H-1
1656  YT(LL,DD)=YT(LL,DD+1)
1658  NEXT DD
1660  YT(LL,TOR(LL)/H)=Y(LL)
1662  NEXT LL
1664  RETURN
1800  REM LPRINT PROGRAMS 2
1802  IF PR=0 THEN 800
1804  GOSUB 1900
1806  LPRINT
1808  LPRINT"Time"
1810  FOR I=1 TO NP
```

```
1812  LPRINT TAB(9+(I-1)*(60/NP));"Y(";I;")";
1814  NEXT I
1816  LPRINT
1818  LPRINT
1820  IF BN<>NB THEN 1848
1822  IF PR=0 THEN 816
1824  DT=INT(T*1000+.5)/1000
1826  LPRINT DT;
1832  REM PRINT"                "
1834  FOR LL=1 TO NP
1838  OTT=INT(OT(LL)*1000+.5)/1000
1840  LPRINT TAB(9+(LL-1)*(60/NP));OTT;
1842  NEXT LL
1844  LPRINT
1846  NB=0
1848  NB=NB+1
1850  RETURN
1900  REM print head
1902  LPRINT"The Number of State Variables";N
1904  LPRINT"The Number of Input Variables";NR
1906  LPRINT"The Number of Output Variables";NP
1908  LPRINT"The Integrate Step Length";H
1910  LPRINT"The Integrate End Time";TE
1912  LPRINT"Interval of Printing Points";BN
1914  LPRINT
1916  LPRINT"MATRIX A:";N;"x";N
1918  FOR I=1 TO N
1920  FOR J=1 TO N
1922  LPRINT TAB(J*8+3);A(I,J):REM STATE MATRIX ELEMENTS
1924  NEXT J
1926  NEXT I
1928  LPRINT"MATRIX B:";N;"x";NR
1930  FOR I=1 TO N
1932  FOR J=1 TO NR
1934  LPRINT TAB(J*8+3);B(I,J):REM INPUT MATRIX ELEMENTS
1936  NEXT J
1938  NEXT I
1940  LPRINT"MATRIX C:";NP;"x";N
1942  FOR I=1 TO NP
1944  FOR J=1 TO N
```

```
1946  LPRINT TAB(J * 8+3);C(I,J):REM OUTPUT MATRIX ELEMENTS
1948  NEXT J
1950  NEXT I
1952  LPRINT"MATRIX D:";NP;"x";NR
1954  FOR I=1 TO NP
1956  FOR J=1 TO NR
1958  LPRINT TAB(J * 8+3);D(I,J):REM OUTPUT MATRIX ELEMENTS 2
1960  NEXT J
1962  NEXT I
1964  IF TR=1 THEN 1968
1966  RETURN
1968  LPRINT"MATRIX A (DEAD TIME):";N;"x";N
1970  FOR I=1 TO N
1972  FOR J=1 TO N
1974  LPRINT TAB(J * 8+3);AT(I,J):REM STATE MATRIX ELEMENTS (DEAD
TIME)
1976  NEXT J
1978  NEXT I
1980  LPRINT"MATRIX C (DEAD TIME):";NP;"x";N
1982  FOR I=1 TO NP
1984  FOR J=1 TO N
1986  LPRINT TAB(J * 8+3);CT(I,J):REM OUTPUT MATRIX ELEMENTS
(DEAD TIME)
1988  NEXT J
1990  NEXT I
1992  RETURN
2000  REM DEV. EQUATION 2
2010  FOR II=1 TO N
2020  F(II)=0
2030  FOR JJ=1 TO N
2040  F(II)=F(II)+A(II,JJ) * Y(JJ)
2050  NEXT JJ
2060  FOR JJ=1 TO NR
2070  F(II)=F(II)+B(II,JJ) * UU(JJ)
2080  NEXT JJ
2090  NEXT II
2100  IF TR=1 THEN GOSUB 1600
2110  RETURN
```

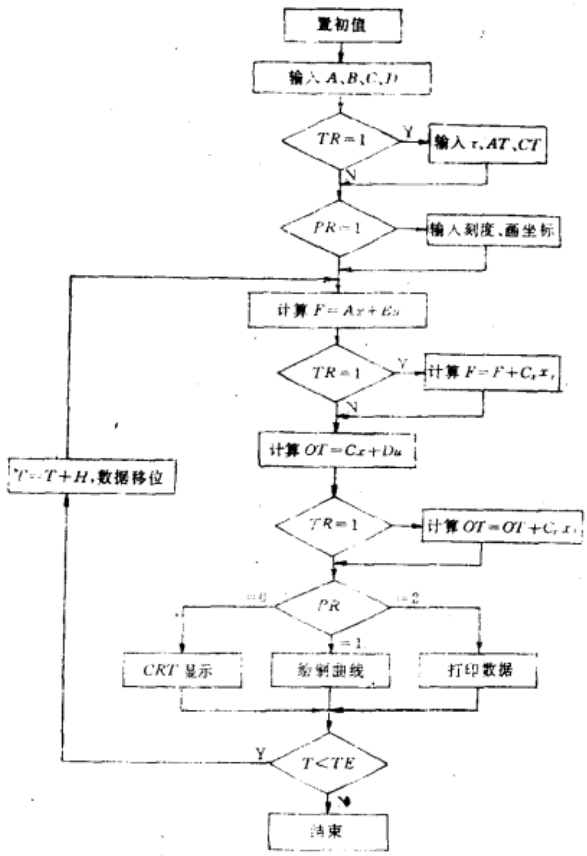


图 2－18　数字仿真程序框图

〔例 2－11〕测取单位阶跃输入下，一阶加纯滞后环节的输出响应特性。

设环节传递函数是 $G(s)=\dfrac{Ke^{-\tau s}}{Ts+1}$。输出信号取有纯滞后及不含纯滞后两项。列出状态方程：

$$\dot{x}=-\frac{1}{T}x+u$$

输出方程：$\begin{pmatrix}y_1\\y_2\end{pmatrix}=\begin{bmatrix}\dfrac{K}{T}\\0\end{bmatrix}x+\begin{bmatrix}0\\\dfrac{K}{T}\end{bmatrix}x_\tau$

当 $K=1$；$T=5$；$\tau=2$ 时，输入数据程序段如下：

```
5000  DATA  0, 1, -0.2, 1, 0.2, 0, 0, 0
5010  DATA  2, 0, 0, 0.2
```

运行后，在显示的? 后打入下列数据：

```
The Number of State Variables   ? 1
The Number of Input Variables   ? 1
The Number of Output Variables  ? 2
The Integrate Step Length       ? .05
The Integrate End Time          ? 5
```

Dead Time Exist (Y or N) ? Y
Interval of Printing Points ? 10
0: CRT 1: DRAW 2: PRINT
Select Output Mode (0, 1, or 2) ? 0

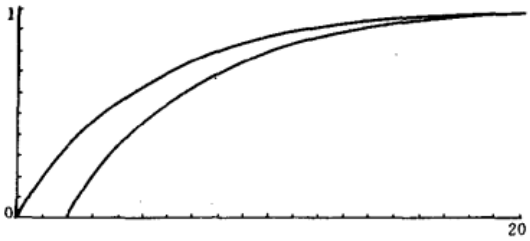


图 2-19 阶跃响应曲线

选择输出方式中，打入 0，则在 CRT 上显示数据，打入 1 则显示如图 2-19。

图形显示前要输入纵坐标刻度范围。本例中，RMAX=1；RMIN=0。下部有 5 行数据区，用于显示有关数据。在ALL9P软件的支持下，即可打印出上述的图形(在九针打印机上)。

如果选择输出方式时，打入 2。则根据打印间隔（本例为 10 点，即按 10×积分步长的时间）打印出如下数据：

The Number of State Variables 1
The Number of Input Variables 1
The Number of Output Variables 2
The Integrate Step Length .05
The Integrate End Time 5
Interval of Printing Points 10
MATRIX A: 1×1
-.2
MATRIX B: 1×1
1
MATRIX C: 2×1
.2
0
MATRIX D: 2×1
0
0
MATRIX A (DEAD TIME): 1×1
0
MATRIX C (DEAD TIME): 2×1
0
.2

Time	Y (1)	Y (2)
0	0	0
.5	.095	0
1	.181	0
1.5	.259	0
2	.33	0

2.5	.393	.095
3	.451	.181
3.5	.5030001	.259
4	.551	.33
4.5	.593	.393
5	.632	.451

打印出来的数据中，Y（1）、Y（2）分别是不含纯滞后、含纯滞后时的输出值。应该指出，打印数据时不需要 ALL9P 软件的支持。

〔例 2－12〕已知对象是 $G(s)=\frac{Ke^{-\tau s}}{Ts+1}$，采用调节器 $G_c(s)=K_c\left(1+\frac{1}{T_i s}\right)$。求取设定值阶跃变化时的闭环输出响应曲线。

根据例 2－10 列出状态方程与输出方程：

$$\dot{x}_1=-\frac{K}{T}x_{2\tau}+R-F$$

$$\dot{x}_2=\frac{K_c}{T_i}x_1-\frac{1}{T}x_2-\frac{K_cK}{T}x_{2\tau}+K_cR-K_cF$$

$$y=F+\frac{K}{T}x_{2\tau}$$

列出相应系数矩阵是：

$$A=\begin{bmatrix}0 & 0\\ \frac{K_c}{T_i} & -\frac{1}{T}\end{bmatrix};B=\begin{bmatrix}1 & -1\\ K_c & -K_c\end{bmatrix};C=\begin{bmatrix}\frac{K_c}{T_i} & 0\\ 0 & 0\end{bmatrix};D=\begin{bmatrix}K_c & -K_c\\ 0 & 0\end{bmatrix}$$

$$A_\tau=\begin{bmatrix}0 & -\frac{K}{T}\\ 0 & -\frac{KK_c}{T}\end{bmatrix};C_\tau=\begin{bmatrix}0 & -\frac{K_cK}{T}\\ 0 & \frac{K}{T}\end{bmatrix}$$

为便于比较，在程序中，把有关数据用 K_c、T_i、K、T、τ（TOR）表示。并在相应曲线下面分别列出。图 2－20 是不同 K_c 时的响应曲线图。不同 T_i 时的响应曲线如图 2－21 所示。τ（TOR）对输出响应的影响如图 2－22 所示。

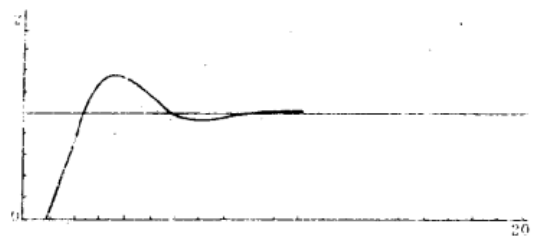
〔例 2－13〕对上述系统，用哈明算法与龙格—库塔算法计算调节器输出与对象输出。

采用哈明算法时，除了在 38 句加入有关数组的定维外，还需用哈明算法子程序代替龙格—库塔算法子程序。

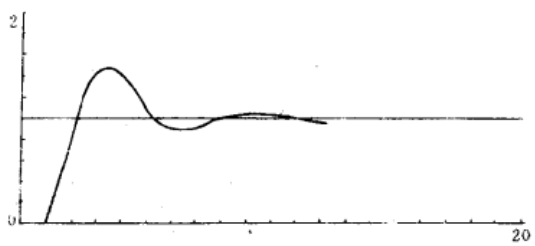
哈明算法的计算结果如下：

Time	Y（1）	Y（2）
0	.5	0
1	1	0
2	1.5	0
3	1.914	.034
4	2.123	.136
5	2.167	.28
6	2.054	.455

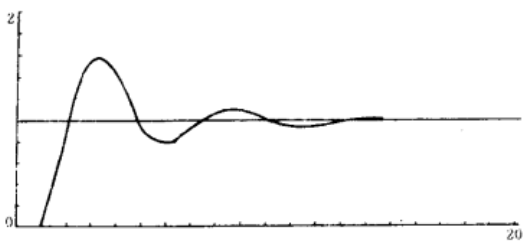
7	1.837	.637
8	1.559	.806
9	1.27	.949
10	1.008	1.055
11	.799	1.122
12	.657	1.151
13	.585	1.149
14	.575	1.125
15	.614	1.087
16	.685	1.045
17	.77	1.005
18	.855	.973
19	.928	.951
20	.982	.94



$Kc=7\ K=0.5\ Ti=3.3\ TOR=1\ T=5$

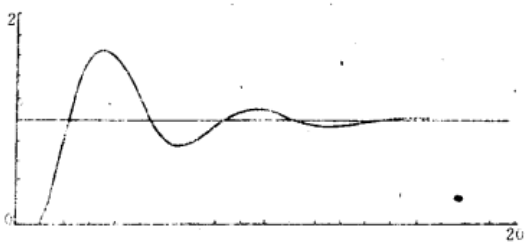


$Kc=8\ K=0.5\ Ti=3.3\ TOR=1\ T=5$

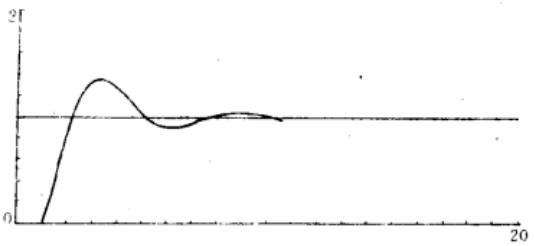


$Kc=9\ K=0.5\ Ti=3.3\ TOR=1\ T=5$

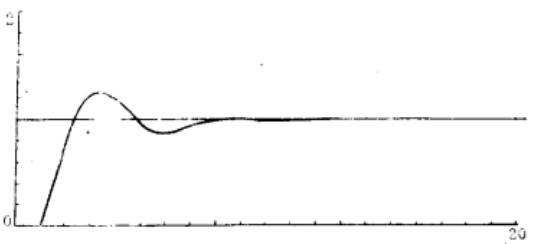
图 2－20　不同 K_c 时的闭环输出响应



$Kc=8\ K=0.5\ Ti=2.3\ TOR=1\ T=5$



$Kc=8\ K=0.5\ Ti=4.3\ TOR=1\ T=5$



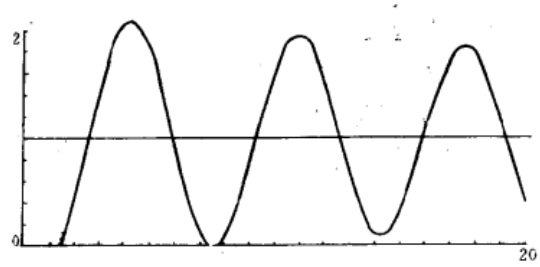
$Kc=8\ K=0.5\ Ti=6.3\ TOR=1\ T=5$

图 2－21　不同 T_i 时的闭环输出响应

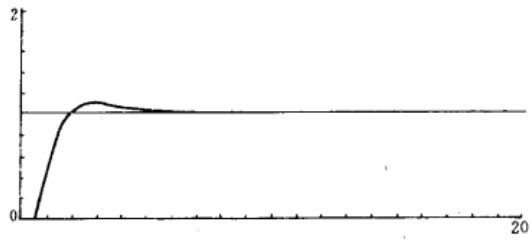
龙格—库塔算法的结果如下：

Time	Y (1)	Y (2)
0	.5	0
1	1	0
2	1.5	0
3	1.913	.034
4	2.12	.136
5	2.16	.28
6	2.046	.454
7	1.829	.634
8	1.556	.801
9	1.273	.941
10	1.018	1.045
11	.817	1.11
12	.681	1.139
13	.612	1.138
14	.603	1.116
15	.638	1.082
16	.703	1.043
17	.781	1.007
18	.858	.977
19	.923	.958
20	.972	.948

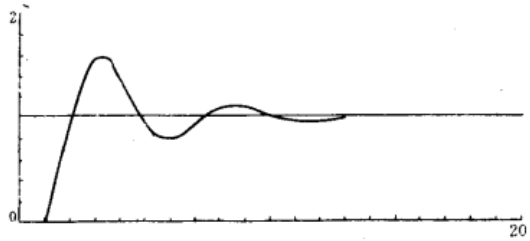
相应的曲线示于图 2－23 及图 2－24。可以看到两者的结果十分接近。



$Kc=9\ K=0.5\ Ti=3.3\ TOR=1.5\ T=5$



$Kc=9\ K=0.5\ Ti=3.3\ TOR=0.5\ T=5$



$Kc=9\ K=0.5\ Ti=3.3\ TOR=1\ T=5$

图 2－22　不同纯滞后时的闭环输出响应

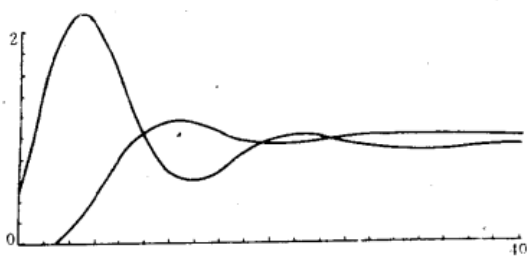


图 2－23　哈明法计算结果

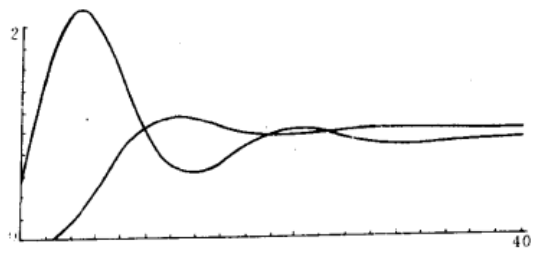


图 2－24　龙格—库塔法计算结果

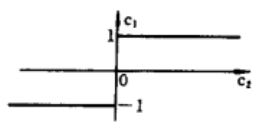
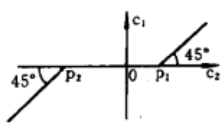
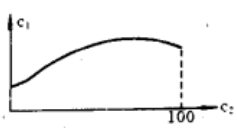
2.4　CSS 数字仿真程序包

本节介绍的 CSS 数字仿真程序包是 1981 年美国宾州大学 K.A.Fegley 教授来华讲学时带来的。由于 CSS 数字仿真程序包便于掌握，容易使用，所以得到较广泛的使用。国内有些单位改成 FORTRAN 语言的 ZFX 程序包，并增加了部分功能。本节主要介绍 CSS 数字仿真程序包。

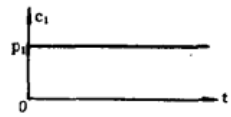
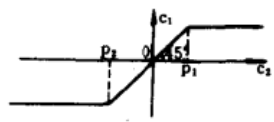
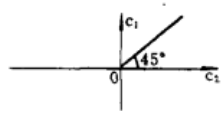
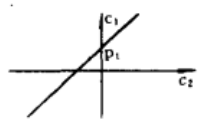
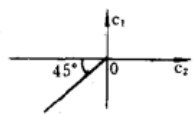
2.4.1 CSS 程序包的基本功能

CSS 程序包是面向框图或结构图的连续系统数字仿真程序包。程序功能块采用模块结构。可对最大容量为 75 个模块的连续系统进行数字仿真，被仿真系统最高阶数为 25 阶。采用 BASIC 语言编写。CSS 程序包有丰富的功能模块可供选用，还留有自定义子程序段，供用户自已编制模块。提供的功能模块如下表：

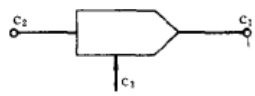
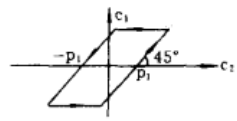
表 2－1 CSS 程序包功能模块一览表

类型号	名 称	功能及使用说明
1	加法器	C1 = ±C2 ± C3 ± C4
2	反 号	C1 = －C2
3	除法器	$C1=\frac{C2}{C3}$
4	反正切	C1 = ATN (C2)　　(C1 = $\tan^{-1}$ (C2))
5	继电非线性	$C1=\begin{cases}1 & C2>0\\0 & C2=0\\-1 & C2<0\end{cases}$ 
6	余 弦	C1 = COS (C2)
7	死区非线性	$C1=\begin{cases}C2-P1 & C2>P1\\0 & P2\leqslant C2\leqslant P1\\C2-P2 & C2<P2\end{cases}$ 
8	指 数	C1 = e^{C2}　　(C1 = EXP (C2))
9	函数发生器	C1 = F (C2)  连接语句：I6, 9, C2, 0, 0 参数语句：I6, N, 1, 0 $X_1, X_2, \cdots, X_N$ $Y_1, Y_2 \cdots, Y_N$
10	增 益	C1 = P1 * C2
11	开 方	C1 = SQR (C2)　　(C1 = $\sqrt{C2}$)

续表

类型号	名　称	功能及使用说明
12	积　分	$C1 = P1 + \int (C2 + C3 * P2 + C4 * P3)dt$　对三个输入端的值加权求和，然后求积分，再加入初值。参数语句 I6，P1，P2，P3
13	随机数发生器	C1 是±1 间的均匀分布随机数
14	阶跃输入	$C1 = \begin{cases} P1 & t \geqslant 0 \\ 0 & t < 0 \end{cases}$ 
15	饱和非线性	$C1 = \begin{cases} P1 & C2 \geqslant P1 \\ C2 & P1 < C2 < P2 \\ P2 & C2 \leqslant P2 \end{cases}$ 
16	绝对值	C1 = ABS（C2）　　（C1 = \|C2\|）
17	负削波	$C1 = \begin{cases} C2 & C2 \geqslant 0 \\ 0 & C2 < 0 \end{cases}$ 
18	零位偏置	C1 = C2 + P1 
19	正削波	$C1 = \begin{cases} 0 & C2 \geqslant 0 \\ C2 & C2 < 0 \end{cases}$ 
20	比较控制	C2≤C3　运行 C2≥C3　停止
21	开　关	$C1 = \begin{cases} C3 & C2 \geqslant 0 \\ C4 & C2 < 0 \end{cases}$
22	正　弦	C1 = SIN（C2）　　（C1 = sin C2）
23	脉冲发生器	C2＜0，C1 = 0 C2≥0，C1 为周期 P1 的单位脉冲
24	半步长延迟器	t≤Δt/2，C1 = P1　　Δt：步长 t＞Δt/2，C1 = C2（t－Δt/2）
25	计算编号寄存器	C（76）= 0 时，寄存计算编号为 N1
26	加权和	C1 = P1 * C2 + P2 * C3 + P3 * C4
27	乘法器	C1 = C2 * C3

续表

类型号	名　称	功能及使用说明
28	条件转移控制	
29	零阶保持器	C3 是负脉冲时，采样 C3 是正脉冲时，采样保持
30	对　数	P1 = 0 时 C1 = ln（C2） P1 = 1 时 C1 = lg（C2）
31	锯齿波发生器	$C1 = C2 - P1 * INT\left(\frac{C2}{P1}\right)$　　INT（I）为 I 的取整函数
32	滞　环 （用户自定义）	
33～38	（用户自定义）	

本节的程序清单中提拱了一个自定义功能模块的程序。CSS 程序包有 7 个可供用户自定义的模块部位。应该指出，CSS 程序包是一个早期的仿真程序包，它对于象一阶、二阶惯性环节、纯滞后环节等都无现成的模块可供选用，而且所用的数值积分法只有四阶龙格—库塔法。但作为程序包的基本结构的学习，还是可以采用的。

2.4.2　CSS 程序包清单及变量、数组说明

CSS 程序包清单如后。为便于阅读，现把该程序包中有关变量、数组说明如下：

C1：功能模块输出变量。

C2：功能模块第一输入端变量。

C3：功能模块第二输入端变量。

C4：功能模块第三输入端变量。

D3：积分步长。

D4：积分半步长。

D5：四分之一积分步长。

A1：积分功能模块输出值控制变量。

A2：打印点数累加指示变量。

K8：打印控制变量。

K1～K6：输出打印变量的功能模块自然编号。

M2：打印步数。

M3：计算积分步数累计变量。

N1：功能模块指针。

N8：被仿真系统功能模块总数。

P1～P3：功能模块第一到第三输入端的增益。

S9：打印工作方式控制变量。

T1：打印间隔。

T2：仿真总时间 T2。

C（76）：功能模块输出值存放数组。最多可存放 75 个输出值。C（76）存放时钟，指示当前时刻。

P（75，3）：功能模块三个输入端参数存放数组。

N（76）：功能模块计算顺序编号存放数组。

U（75，5）：功能模块特征值存放数组。每个功能模块有五个特征值。U（I，1）是存放模块的类型号，U(I,2)、U(I,3)、U(I,4) 是存放连接该模块的三个输入端的功能模块编号，U（I，5）是存放功能模块分类后的编号。

V（25）：积分功能模块统计编号存放数组。

F（3，11）：函数发生器功能模块的数值的存放数组。最多 3 个函数发生器，每个函数发生器有 11 个值。

D（25）：单位时间保持器统计编号存放数组。

Y（25）：积分功能模块输出值存放数组。

T（25）：积分功能模块等效总输入 T。

W（25）：积分计算开始时，存放上步各积分功能模块输出值的数组。

B（150，6）：绘图数据存放数组。最多存放 6 条曲线。每条曲线 150 个点。

Q（6，2）：用于绘图打印时存放数据值大小的顺序及打印标记符号。

E（25，4）：龙格—库塔算法的四个系数。

Z（4）：绘图时，打印出幅值坐标标度值。

I6：功能模块类型号。U（I，1）= I6。

J、K、L：送入的功能模块 I6 的三个输入端相应功能模块的自然编号。

U（I，2）= J；U（I，3）= K；U（I，4）= L。

Z1：绘制曲线的条数。

A 与 Z0：绘制曲线的起点与终点。

```
1   REM Continuous System Simulation(CSS Program)
100   DIM C(76),P(75,3),N(76),U(76,5),V(25),F(3,11)
110   DIM D(25),Y(25),T(25),W(25),B(150,6),Q(6,2)
120   DIM E(25,4),Z(4)
130   LPRINT"Continuous System Simulation"
140   LPRINT
150   READ I,I6,J,K,L
160   IF I<=0 THEN 220
170   U(I,1)=I6
180   U(I,2)=J
190   U(I,3)=K
200   U(I,4)=L
210   GOTO 150
220   FOR I=1 TO 75
230   IF U(I,1)<0 THEN 500
240   IF U(I,1)<39 THEN 530
```

```
500  PRINT"I11egal Type, Block(";I;")Enter Type"
510  INPUT U(I,1)
520  GOTO 230
530  NEXT I
540  N3=0:N4=0:N9=0:N2=0
550  FOR I=1 TO 75
560  I6=U(I,1)
570  IF I6<>24 THEN 610
580  N4=N4+1
590  D(N4)=I
600  GOTO 730
610  IF I6<>12 THEN 660
620  N9=N9+1
630  V(N9)=I
640  U(I,5)=N9
650  GOTO 730
660  IF I6<>14 THEN 700
670  N3=N3+1
680  N(N3)=I
690  GOTO 740
700  IF I6<>9 THEN 730
710  N2=N2+1
720  U(I,5)=N2
730  U(I,1)=-I6
740  NEXT I
760  N(N3+1)=76
770  N8=N3+1
780  FOR I=1 TO 75
790  IF U(I,1)>=0 THEN 1050
800  FOR K=2 TO 4
810  L1=ABS(U(I,K))
820  IF L1=0 THEN 1010
825  IF N4=0 THEN 860
830  FOR J=1 TO N4
840  IF L1=D(J) THEN 1000
850  NEXT J
860  IF N9>0 THEN 930
870  LPRINT"Network Must Contain At Least"
875  LPRINT "One Integator"
880  LPRINT"Recheck Data Statements and Restart"
```

```
890  END
930  FOR J=1 TO N9
940  IF L1=V(J) THEN 1000
950  NEXT J
960  FOR J=1 TO N8
970  IF L1=N(J) THEN 1000
980  NEXT J
990  GOTO 1050
1000  NEXT K
1010  N8=N8+1
1020  N(N8)=I
1030  U(I,1)=-U(I,1)
1040  GOTO 780
1050  NEXT I
1060  I4=0
1070  FOR I=1 TO 75
1080  IF U(I,1)>=0 THEN 1120
1090  I4=2
1100  U(I,I)=-U(I,1)
1110  LPRINT"Sort Failure, Block";I
1120  NEXT I
1130  IF I4>0 THEN 890
1140  IF N2=0 THEN 1320
1180  FOR I=1 TO 75
1190  IF U(I,1)<>9 THEN 1250
1200  J=U(I,5)
1220  FOR J1=1 TO 11
1230  READ F(J,J1)
1240  NEXT J1
1250  NEXT I
1320  READ I,C(1),C(2),C(3)
1350  IF I=0 THEN 1405
1360  FOR J=1 TO 3
1370  P(I,J)=C(J)
1380  NEXT J
1390  GOTO 1320
1405  READ D3,T2,T1,K1,K2,K3,K4,K5,K6
1450  D4=D3/2
1455  D5=D3/4
1460  M5=T1/D3
```

```
1480  K8 = 1
2430  S9 = 0
2450  FOR J = 1 TO 25
2452  E(J, 1) = 0
2454  E(J, 2) = 0
2456  E(J, 3) = 0
2458  E(J, 4) = 0
2470  NEXT J
2475  A2 = 0
2480  A1 = 1
2490  FOR N1 = 1 TO N8
2500  I = N(N1)
2510  IF I = 76 THEN 2540
2520  C(I) = P(I, 1)
2530  GOTO 2550
2540  C(I) = 0
2550  NEXT N1
2570  FOR I3 = 1 TO N9
2580  Y(I3)P(V(I3), 1)
2590  NEXT I3
2610  PRINT"Do you Want a Configuration Recap?"
2620  INPUT I$
2630  IF I$ = "N"OR I$ = "n"THEN 2770
2640  LPRINT
2650  LPRINT "Block"; TAB(8)"Type"; TAB(18)"IP - 1";
2655  LPRINT TAB(34)"IP - 2"TAB(51)"IP - 3"
2660  LPRINT TAB(17)"Par - 1"TAB(33)"Par - 2"TAB(50)"Par - 3"
2665  LPRINT:LPRINT
2690  FOR I = 1 TO 75
2700  IF U(I, 1) = 0 THEN 2765
2710  LPRINT I; TAB(8); U(I, 1);
2720  FOR 0 = 1 TO 3
2730  LPRINT TAB(0 * 16 - 1); U(I, 0 + 1)
2740  NEXT 0
2745  LPRINT
2750  FOR 0 = 1 TO 3
2755  LPRINT TAB(0 * 16 - 1); P(I, 0);
2760  NEXT 0
2763  LPRINT
2765  NEXT I
```

```
2770  LPRINT:LPRINT:LPRINT
2775  K(1)=K1:K(2)=K2:K(3)=K3:K(4)=K4:K(5)=K5:K(6)=K6
2780  LPRINT TAB(4);"Time";
2785  FOR 0=1 TO 3
2790  LPRINT TAB(0*15+3);"Output-";K(0);
2795  NEXT 0
2800  LPRINT
2805  FOR 0=1 TO 3
2810  LPRINT TAB(0*15+3);"Output-";K(0+3)
2815  NEXT 0
2820  LPRINT:LPRINT:LPRINT
2830  GOTO 3280
2832  LPRINT C(76)TAB(19)C(K1)TAB(33)C(K2)TAB(48)C(K3)
2834  LPRINT TAB(19)C(K4)TAB(33)C(K5)TAB(48)C(K6)
2841  A2=A2+1
2842  B(A2,1)=C(K1)
2843  B(A2,2)=C(K2)
2844  B(A2,3)=C(K3)
2845  B(A2,4)=C(K4)
2846  B(A2,5)=C(K5)
2847  B(A2,6)=C(K6)
2850  LPRINT
2860  M2=0
2870  M2=M2+1
2880  M3=0
2890  K8=2
2900  FOR M=1 TO N9
2910  ON A1 GOTO 2920,2920,2930,2950,9999
2920  W(M)=Y(M)
2930  Y(M)=W(M)+D4*T(M)
2940  GOTO 2960
2950  Y(M)=W(M)+D3*T(M)
2960  NEXT M
2990  IF A1<>3 THEN 3012
3000  M3=M3+1
3010  GOTO 3280
3012  C(76)=C(76)+D4
3014  GOTO 3280
3020  K8=3
3030  FOR M=1 TO N9
```

```
3040  Y(M) = W(M) + D3 * ((E(M, 1) + E(M, 4))/6 + (E(M, 2) + E(M, 3))/3)
3050  NEXT M
3080  GOTO 3280
3090  IF M3 < M5 - .00005 THEN 2890
3100  LPRINT C(76)TAB(19)C(K1)TAB(33)C(K2)TAB(48)C(K3)
3110  LPRINT TAB(19)C(K4)TAB(33)C(K5)TAB(48)C(K6)
3111  A2 = A2 + 1
3112  B(A2, 1) = C(K1)
3113  B(A2, 2) = C(K2)
3114  B(A2, 3) = C(K3)
3115  B(A2, 4) = C(K4)
3116  B(A2, 5) = C(K5)
3117  B(A2, 6) = C(K6)
3118  LPRINT
3130  IF C(76) < T2 THEN 2870
3140  LPRINT
3150  LPRINT
3160  LPRINT "Enter Code:"
3170  IF S9 = 1 THEN 3230
3180  LPRINT"1 Continue Run"
3190  LPRINT"2 Modify Initial Conditions"
3200  LPRINT"3 Modify Timing and Output Block Number"
3205  LPRINT"4 Restart Run"
3210  LPRINT"5 End of Run"
3215  LPRINT"6 Plot"
3220  PRINT "MO = ";
3230  INPUT MO
3240  S9 = 1
3250  ON MO GOTO 5310, 5370, 5450, 5520, 9999, 5560
3260  S9 = 0
3270  GOTO 3150
3280  FOR I3 = 1 TO N9
3290  I = V(I3)
3300  C(I) = Y(I3)
3310  NEXT I3
3320  N1 = N3 + 2
3330  I = N(N1)
3340  I6 = U(I, 1)
3350  J = U(I, 2)
3360  K = U(I, 3)
```

```
3370  L=U(I,4)
3380  IF J=0 THEN 3420
3390  J1=ABS(J)
3400  C2=C(J1)
3410  GOTO 3430
3420  J1=0
3422  C2=0
3430  IF K=0 THEN 3470
3440  K7=ABS(K)
3450  C3=C(K7)
3460  GOTO 3480
3470  K7=0
3472  C3=0
3480  IF L=0 THEN 3520
3490  L2=ABS(L)
3500  C4=C(L2)
3510  GOTO 3530
3520  L2=0
3522  C4=0
3530  P1=P(I,1)
3540  P2=P(I,2)
3550  IF I6>=20 THEN 3600
3560  IF I6>=11 THEN 3580
3570  ON I6 GOTO 3650,5150,3780,3800,3820,3850,3870,3940,3960,4130
3580  I5=I6-10
3590  ON I5 GOTO 4150,4170,4260,5210,4310,4360,4380,4390,4410
3600  I5=I6-19
3610  IF I5>=11 THEN 3630
3620  ON I5 GOTO 4420,4430,4480,4500,4580,4650,4680,4740,4760,4810
3630  I5=I5-10
3640  ON I5 GOTO 4880,4920,4940,4960,4980,5000,5020,5040,5060
3650  IF J>0 THEN 3680
3660  C1=-C2
3670  GOTO 3690
3680  C1=C2
3690  ON SGN(K)+2 GOTO 3700,5200,3720
3700  C1=C1-C3
3710  GOTO 3730
3720  C1=C1+C3
3730  ON SGN(L)+2 GOTO 3740,5200,3760
```

```
3740  C1 = C1 - C4
3750  GOTO 5200
3760  C1 = C1 + C4
3770  GOTO 5200
3780  C1 = C2/C3
3790  GOTO 5200
3800  C1 = ATN(C2)
3810  GOTO 5200
3820  ON SGN(C2) + 2 GOTO 3830, 5190, 5130
3830  C1 = -1
3840  GOTO 5200
3850  C1 = COS(C2)
3860  GOTO 5200
3870  ON SGN(C2) + 2 GOTO 3920, 5190, 3880
3880  D2 = C2 - P1
3890  IF D2 <= 0 THEN 5190
3900  C1 = D2
3910  GOTO 5200
3920  D2 = C2 - P2
3930  ON SGN(D2) + 2 GOTO 3900, 5190, 5190
3940  C1 = EXP(C2)
3950  GOTO 5200
3960  N6 = U(I, 5)
3970  IF N6 > 0 THEN 4000
3980  C1 = 0
3990  GOTO 5200
4000  IF C2 > 0 THEN 4030
4010  C1 = F(N6, 1)
4020  GOTO 5200
4030  IF C2 < 100 THEN 4060
4040  C1 = F(N6, 11)
4050  GOTO 5200
4060  FOR M = 1 TO 10
4070  V1 = 10 * M
4080  IF V1 >= C2 THEN 4100
4090  NEXT M
4100  F1 = .1 * (C2 - V1 + 10)
4110  C1 = F(N6, M) + F1 * (F(N6, M + 1) - F(N6, M))
4120  GOTO 5200
4130  C1 = P1 * C2
```

```
4140  GOTO 5200
4150  C1 = SQR(C2)
4160  GOTO 5200
4170  I3 = U(I, 5)
4180  D1 = C2
4190  ON SGN(K) + 2 GOTO 9999, 4210, 4200
4200  D1 = D1 + P2 * C3
4210  ON SGN(L) + 2 GOTO 9999, 4230, 4220
4220  D1 = D1 + P(I, 3) * C4
4230  T(I3) = D1
4240  E(I3, A1) = T(I3)
4250  GOTO 5210
4260  ON K8 GOTO 4270, 4270, 5210
4270  I2 = - (I2 * 259)
4280  C1 = I2/(32767 * 10^-7)
4290  C1 = 2 * (RND(I2) - .5)
4300  GOTO 5200
4310  D2 = C1 - P1
4320  IF D2 >= 0 THEN 5090
4330  D2 = C2 - P2
4340  IF D2 >= 0 THEN 5170
4350  GOTO 5110
4360  C1 = ABS(C2)
4370  GOTO 5200
4380  ON SGN(C2) + 2 GOTO 5190, 5190, 5170
4390  C1 = C2 + P1
4400  GOTO 5200
4410  ON SGN(C2) + 2 GOTO 5170, 5190, 5190
4420  ON SGN(C2 - C3) + 2 GOTO 5210, 5210, 3160
4430  IF C2 < 0 THEN 4460
4440  C1 = C3
4450  GOTO 5200
4460  C1 = C4
4470  GOTO 5200
4480  C1 = SIN(C2)
4490  GOTO 5200
4500  ON SGN(C(76)) + 2 GOTO 9999, 4510, 4530
4510  P(I, 2) = - P1
4520  ON SGN(C2) + 2 GOTO 5190, 5130, 5130
4530  ON SGN(C2) + 2 GOTO 5190, 5130, 4540
```

```
4540  P(I,2)=P2+D3/4
4550  IF P(I,2)<0 THEN 5190
4560  P(I,2)=-P1
4570  GOTO 5130
4580  ON SGN(C(76))+2 GOTO 9999,4590,4610
4590  C1=P1
4600  GOTO 4630
4610  IF C(76)<D4 THEN 4590
4620  C1=P2
4630  P(I,2)=C2
4640  GOTO 5200
4650  ON SGN(C(76))+2 GOTO 9999,4660,5210
4660  U(I,5)=N1
4670  GOTO 5210
4680  C1=C2*P1
4690  ON SGN(K)+2 GOTO 9999,5200,4700
4700  C1=C1+C3*P2
4710  ON SGN(L)+2 GOTO 9999,5200,4720
4720  C1=C1+C4*P(I,3)
4730  GOTO 5200
4740  C1=C2*C3
4750  GOTO 5200
4760  R1=ABS(C3-C2)/ABS(C2)
4770  IF R1<=P1 THEN 5170
4780  C(K)=(1-P2)*C2+P2*C3
4790  N1=U(K,5)
4800  GOTO 3330
4810  ON SGN(C(76))+2 GOTO 9999,4820,4840
4820  P(I,2)=P1
4830  P2=P1
4840  ON SGN(C3)+2 GOTO 5190,5110,4850
4850  P(I,2)=C2
4860  P2=C2
4870  GOTO 5110
4880  C1=LOG(C2)
4890  IF P1=0 THEN 5200
4900  C1=.434492*C1
4910  GOTO 5200
4920  C1=C2-INT(C2/P1)*P1
4930  GOTO 5200
```

```
4940  ON SGN(C2 - P2) + 2 GOTO 4941, 4942, 4946
4941  ON SGN(P(I, 3) - C2 - P1) + 2 GOTO 4942, 4944, 4944
4942  C1 = P(I, 3)
4943  GOTO 4948
4944  C1 = C2 + P1
4945  GOTO 4948
4946  ON SGN(P(I, 3) - C2 + P1) + 2 GOTO 4947, 4947, 4942
4947  C1 = C2 - P1
4948  P(I, 2) = C2
4949  P(I, 3) = C1
4950  GOTO 5200
4960  REM
4970  GOTO 5070
4980  REM
4990  GOTO 5070
5000  REM
5010  GOTO 5070
5020  REM
5030  GOTO 5070
5040  REM
5050  GOTO 5070
5060  REM
5070  LPRINT"User Subroutine"; I6; "Incorrect or Not Present"
5080  END
5090  C1 = P1
5100  GOTO 5200
5110  C1 = P2
5120  GOTO 5200
5130  C1 = 1
5140  GOTO 5200
5150  C1 = - C2
5160  GOTO 5200
5170  C1 = C2
5180  GOTO 5200
5190  C1 = 0
5200  C(I) = C1
5210  IF N1 = N8 THEN 5240
5220  N1 = N1 + 1
5230  GOTO 3330
5240  A1 = A1 + 1
```

```
5250  ON K8 GOTO 5260, 5280, 3090
5260  A1 = 2
5270  GOTO 2832
5280  IF A1 < >5 THEN 2900
5290  A1 = 1
5300  GOTO 3020
5310  LPRINT
5320  PRINT"Enter New Total Time"
5330  INPUT T2
5340  LPRINT
5350  GOTO 2870
5360  LPRINT
5370  LPRINT
5380  PRINT"Enter: Block, Par - 1, Par - 2, Par - 3"
5390  INPUT I, J, K, L
5400  IF I = 0 THEN 3150
5410  P(I, 1) = J
5420  P(I, 2) = K
5430  P(I, 3) = L
5440  GOTO 5390
5450  LPRINT
5460  PRINT"Enter: Integrate Interval, Total Time"
5470  PRINT"Print Interval, 0P1, 0P2, 0P3, 0P4, 0P5, 0P6"
5480  INPUT D3, T2, T1, K1, K2, K3, K4, K5, K6
5490  D4 = D3/2
5495  D5 = D3/4
5500  M5 = T1/D3
5510  GOTO 3150
5520  K8 = 1
5530  LPRINT
5540  LPRINT
5550  GOTO 2450
5560  GOSUB 6010
5570  GOTO 3150
6010  GOTO 7070
6020  PRINT"INPUT Block Output To Plot(1 to 6)"
6022  INPUT Z1
6030  IF Z1 < 1 OR Z1 > 6 THEN 6020
6050  IF A2 < 1 OR A2 > 150 THEN 7020
6120  PRINT"Plot Time Index: From, To"
```

```
6130  INPUT A, Z0
6140  IF A = Z0 THEN 7041
6150  IF Z0 > A2 OR A < 1 OR Z0 < A THEN 7000
6180  LPRINT
6190  LPRINT
6200  Q3 = B(A, 1)
6202  Q1 = B(A, 1)
6210  FOR I = A TO Z0
6220  FOR J = 1 TO Z1
6230  IF B(I, J) < = Q1 THEN 6250
6240  Q1 = B(I, J)
6250  IF B(I, J) > = Q3 THEN 6270
6260  Q3 = B(I, J)
6270  NEXT J
6280  NEXT I
6290  IF Q3 < > Q1 THEN 6340
6300  PRINT"y = "; Q3; "Over the Selected Interval"
6310  PRINT
6320  S2 = 1
6330  GOTO 6370
6340  S2 = (Q1 - Q3)/60
6350  LPRINT"Block Output (Ordinate): Each Division = "; S2
6355  LPRINT"Time Scale: Last Set Of Div. = "; T1; "Seconds"
6360  LPRINT
6370  GOSUB 6950
6380  GOSUB 6930
6390  LPRINT
6400  FOR I = A TO Z0
6410  W1 = I/5
6420  IF INT(W1) = W1 THEN 6450
6430  LPRINT".";
6440  GOTO 6460
6450  LPRINT"+";
6460  FOR J = 1 TO Z1
6470  Q2 = INT((B(I, J) - Q3)/S2 + 1.5)
6480  FOR K = 1 TO J
6490  IF Q2 > = Q(K, 1) THEN 6580
6500  IF K > + J THEN 6550
6510  FOR P = J TO K + 1 STEP - 1
6520  Q(P, 1) = Q(P - 1, 1)
```

```
6530  Q(P,2)=Q(P-1,2)
6540  NEXT P
6550  Q(K,1)=Q2
6560  Q(K,2)=J
6570  GOTO 6610
6580  NEXT K
6590  Q(J,1)=Q2
6600  Q(J,2)=J
6610  NEXT J
6620  Q0=1
6630  FOR J=1 TO Z1
6640  Q2=Q(J,1)-Q0
6650  IF Q2<0 THEN 6840
6660  FOR M6=1 TO Q2+1
6670  IF M6=Q2+1 THEN 6700
6680  LPRINT" ";
6690  NEXT M6
6700  Q2=Q(J,2)
6710  ON Q2 GOTO 6740,6760,6780,6800,6800,6820,6720
6720  LPRINT"6";
6730  GOTO 6830
6740  LPRINT"1"
6750  GOTO 6830
6760  LPRINT"2";
6770  GOTO 6830
6780  LPRINT"3";
6790  GOTO 6830
6800  LPRINT"4";
6810  GOTO 6830
6820  LPRINT"5";
6830  Q0=Q(J,1)+1
6840  NEXT J
6850  LPRINT" "
6860  NEXT I
6870  LPRINT
6880  GOSUB 6930
6890  GOSUB 6980
6900  LPRINT
6910  LPRINT
6920  GOTO 6120
```

```
6930  LPRINT"I...+...+...I...+...+...I...+...+...I...+...+...I"
6940  RETURN
6950  FOR J=1 TO 3
6960  Z(J)=Q3+15*S2*J
6970  NEXT J
6980  LPRINT Q3;TAB(15);Z(1);TAB(30);Z(2);TAB(45);Z(3);TAB(60);Q1
6990  RETURN
7000  LPRINT"Index Specified Outside Print Interval Range:"
7005  LPRINT"Min=1;Max=";A2
7010  GOTO 6120
7020  LPRINT"Print Interval has Exceeded Permitted Range:"
7030  LPRINT"1<=Print Interval<=150"
7040  GOTO 6120
7041  LPRINT"End of Data-Plot Routine"
7042  LPRINT"Returning to Calling Program"
7044  RETURN
7060  GOTO 6120
7070  LPRINT"*****Data Plot Routine*****"
7080  LPRINT
7090  PRINT"Do You Want Instructions(answer Y or N)"
7100  INPUT Z$
7110  IF Z$="N"OR Z$="n"THEN 6020
7120  LPRINT
7130  LPRINT
7140  LPRINT"This program will plot the ordinate values of one to six"
7150  LPRINT"block outputs, each set of one to six ordinate values is plotted"
7160  LPRINT"on one line of output, the user may specify the number of"
7170  LPRINT"continuous sets of ordinate values(block outputs)to be plotted"
7180  LPRINT"along the abscissa axis"
7190  LPRINT"In the plot, the first block output is represented by the"
7200  LPRINT"character '1', the second by '2', and so on. If two or more"
7210  LPRINT"ordinate values are equal then the plot character for the first"
7220  LPRINT"of the equal values will take precence."
7230  LPRINT"During execution of the program you will be asked to enter"
7240  LPRINT"the lower and the upper time index of the range of set of print"
7250  LPRINT"interval values to be plotted. Where 1<=The lower time index"
7260  LPRINT"and the upper time index is <=M. For example, the print time"
7270  LPRINT"interval is 1 second and total time is 23 seconds, then these"
7280  LPRINT"would produce 24 values every block(inc. output time=0)."
7290  LPRINT"So the plot time index is therefor:1,24."
```

```
7445  LPRINT"* * * * * To Terminate The Plot Routine Enter 0 for The Lower and"
7450  LPRINT"The Upper Time Index(0,0)."
7500  GOTO 6020
9999  END
```

2.4.3 CSS 仿真程序包的应用

为便于应用 CSS 仿真程序包解决实际问题，在使用前要做如下准备工作：

①对各组成环节列写微分方程式。

②采用积分模块及其他模块，组成仿真框图。

③对组成仿真框图的各模块进行编号。

④根据各模块的编号、连接关系、模块参数列写输入数据语句，并送入。

⑤根据仿真要求，列写积分步长、仿真时间、仿真间隔及输出数据的模块号的语句，并送入。

2.4.3.1 用 CSS 模块组成曲型环节

下面用 CSS 仿真程序包提供的模块组成若干曲型环节。

1. 一阶惯性环节

$$G(s) = \frac{K}{Ts + 1}$$

改成微分方程形式

$$\dot{y}(t) = \frac{K}{T}u(t) - \frac{1}{T}y(t)$$

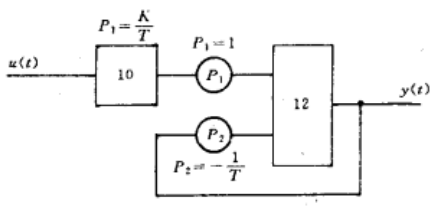


图 2－25　一阶环节的 CSS 组成

这个环节可用两个模块组成。即 10 号模块比例器和 12 号模块积分器。如图 2－25 所示。

图中，比例器的参数 $P = \frac{K}{T}$。积分器的输入端 C2 接比例器输出。输入端 C1 经参数 $P_2 = -\frac{1}{T}$ 输入到积分器。积分器的输出即反馈到 C1 端。

2. 超前滞后环节

$$G(s) = K\frac{1 + T_1 s}{1 + T_2 s}$$

上式可改写成：

$$G(s) = \frac{KT_1}{T_2} + \frac{K\left(1 - \frac{T_1}{T_2}\right)}{1 + T_2 s}$$

因此，它可用比例器、积分器与加法器组成。图 2－26 是由 10、12、26 三个模块组成的连接图。其中，比例器参数 $P = \frac{K\left(1 - \frac{T_1}{T_2}\right)}{T_2}$；积分器的参数 $P_2 = -\frac{1}{T_2}$；加法器参数 $P_1 = K\frac{T_1}{T_2}$；$P_2 = 1$。

3. 二阶振荡环节

$$G(s) = \frac{K}{s^2 + 2\xi\omega s + \omega^2}$$

写成时域方程：

$$\ddot{y}(t) = Ku(t) - 2\xi\omega\dot{y}(t) - \omega^2 y(t)$$

因此，可用图2－27的二个积分器和一个比例器组成。比例器系数 $P=K$，第一个积分器 $P_1=1$；$P_2=-\omega^2$；$P_3=-2\xi\omega$；第二个积分器 $P_1=1$。

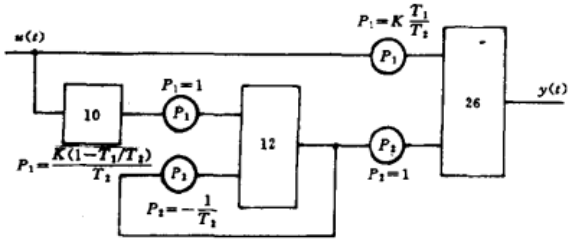


图 2－26　超前滞后环节的CSS组成

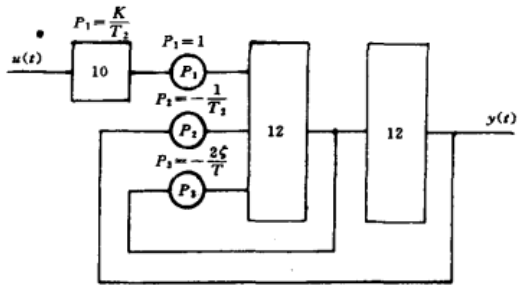


图 2－27　二阶振荡环节的CSS组成

2.4.3.2　用户自编运算功能模块子程序

CSS仿真程序包提供了编号从32到38的7个自定义运算功能模块的程序空间。每个模块占19句语句，其中最后一句为GOTO 5200。下面以滞环非线性为例予以说明。采用32号模块存放该子程序。根据CSS规定的语句范围，32号模块从4940语句开始，最后为4959语句。滞环特性如图2－28所示。设输入为 C_2，输出为 C_1，由于滞环特性具有记忆功能，设上次输入 P_2，上次输出 P_3。则有下列关系：

$C_2-P_2>0$　且 $P_3-(C_2-P_1)\leqslant 0$　则 $C_1=C_2-P_1$

$C_2-P_2>0$　且 $P_3-(C_2-P_1)\geqslant 0$　则 $C_1=P_3$

$C_2-P_2<0$　且 $P_3-(C_2+P_1)<0$　则 $C_1=P_3$

$C_2-P_2<0$　且 $P_3-(C_2+P_1)\geqslant 0$　则 $C_1=C_2+P_1$

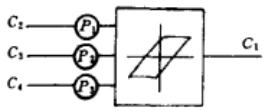


图 2－28　滞环特性

据此，可列出程序清单中4940～4950语句。可以看到最后一句是GOTO 5200。本子程序共用11句语句。

用户可以根据需要编制特定功能模块的子程序，国内有些单位在改进程序包时，就编制了常用典型环节的功能模块，在应用时可以方便地调用。

2.4.3.3　数据输入语句

在系统分解为用CSS的功能模块构成图后，需要对各组成的模块送入有关数据。输入的数据有三类，分别是连接数据、模块参数数据及运行控制参数数据。输入语句格式如下。

1．连接数据语句

首先对组成系统的模块进行编号。然后按下列顺序对每一个模块送入该模块与其他模块的连接关系的数据：

编号，模块类型号，连接关系3个数据

例如一阶惯性环节 $\dfrac{3}{5s+1}$ 用下列语句

4 DATA　1，10，0，0，0

5 DATA　2，12，2，0，0

连接语句最后用5个0结束。

2．参数数据语句

参数数据语句在连接语句后面送入。它是各模块输入端的参数。应注意，积分器的 P_1 表示初值。参数数据格式是：

编号，C_1，C_2，C_3

例如，一阶惯性环节 $\frac{3}{5s+1}$ 的参数语句如下

10 DATA 1，0.6，0，0

11 DATA 2，0，−0.2，0

参数数据语句的结束用4个0表示。

3. 运行控制参数语句

运行控制参数语句包括积分步长、仿真时间、间隔及6个可供输出数据的模块编号。其格式是：

步长，仿真时间，间隔，K_1，K_2，K_3，K_4，K_5，K_6

例如，0.05，1，0.1，2，0，0，0，0，0 表示步长0.05，每隔0.1输出打印模块编号2的输出。仿真时间到1止。

〔例2－4〕已知闭环控制系统如图2－29所示。计算阶跃输入的系统响应。

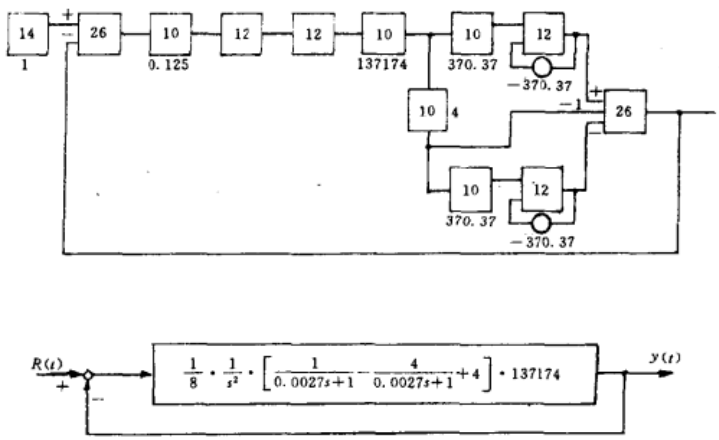


图2－29 闭环系统组成框图

采用CSS的组成模块方法，可以把系统分成12个模块。连接数据语句如下：

```
4   DATA 1,14,0,0,0
6   DATA 2,26,1,12,0
8   DATA 3,10,2,0,0
10  DATA 4,12,3,0,0
12  DATA 5,12,4,0,0
14  DATA 6,10,5,0,0
16  DATA 7,10,6,0,0
18  DATA 8,12,7,8,0
20  DATA 9,10,6,0,0
22  DATA 10,10,9,0,0
24  DATA 11,12,10,11,0
26  DATA 12,26,8,11,9
```

```
28  DATA 0,0,0,0,0
```

其中,28 语句是结束语句。相对应的参数数据语句如下:

```
30  DATA 1,1,0,0
32  DATA 2,1, -1,0
34  DATA 3, .125,0,0
36  DATA 4,0,0,0
38  DATA 5,0,0,0
40  DATA 6,137174,0,0
42  DATA 7,370 .37,0,0
44  DATA 8,0, -370.37,0
46  DATA 9,4,0,0
48  DATA 10,370.37,0,0
50  DATA 11,0, -370.37,0
52  DATA 12,1, -1,1
60  DATA 0,0,0,0
```

同样,结束语句以 4 个 0 结束,如 60 语句。运行参数和控制参数语句如下:

```
90 DATA.0002, .02, .001,12,2,3,4,6,8
```

在 BASIC 运行提示符 OK 出现后,打入 RUN 并回车,则显示:"Do you want a Configuration Recap?"

打入 N 或 n 则转运行程序,否则打印出组态的有关数据,如下所示:

Continuous System Simulation

Block	Type	IP=1 Par-1	IP-2 Par-2	IP-3 Par-3
1	14	0	0	0
		1	0	0
2	26	1	12	0
		1	-1	0
3	10	2	0	0
		.125	0	0
4	12	3	0	0
		0	0	0
5	12	4	0	0
		0	0	0
6	10	5	0	0
		137174	0	0
7	10	6	0	0
		370.37	0	0
8	12	7	8	0
		0	-370.37	0
9	10	6	0	0
		4	0	0
10	10	9	0	0
		370.37	0	0
11	12	10	11	0
		0	-370.37	0
12	26	8	11	9
		1	-1	1

运行结果如下所示：

Time	Output－12 Output－4	Output－2 Output－6	Output－3 Output－8
0	0 0	1 0	.125 0
9.999999E－04	3.121613E－02 1.236679E－04	.9687839 8.527036E－03	.121098 9.64005E－04
1.999999E－03	.1133359 2.400782E－04	.8866641 3.359299E－02	.110833 7.012022E－03
.003	.2308057 3.438642E－04	.7691944 7.381201E－02	9.614929E－02 2.148079E－02
4.000001E－03	.3703389 4.314606E－04	.6296611 .1271889	7.870764E－02 4.613883E－02
4.999999E－03	.5208224 5.008261E－04	.4791777 .1913468	5.989721E－02 8.152165E－02
5.999998E－03	.6731916 5.511791E－04	.3268085 .2637186	4.085106E－02 .1272276
6.999996E－03	.8202806 5.827506E－04	.1797195 .3417018	2.246493E－02 .1821755
7.999994E－03	.9566532 5.965576E－04	4.334677E－02 .4227819	5.418346E－03 .2448248
8.999992E－03	1.078421 5.941977E－04	－7.842088E－02 .5046264	－9.802609E－03 .3133615
9.999991E－03	1.183052 5.77668E－04	－.1830518 .5851507	－2.288148E－02 .3858503
1.099999E－02	1.26918 5.492071E－04	－.2691795 .6625626	－3.364743E－02 .460357
1.199999E－02	1.336416 5.111613E－04	－.3364159 .7353861	－4.205199E－02 .5350429
1.299999E－02	1.385167 4.658741E－04	－.3851674 .8024677	－4.814592E－02 .6082345
1.399998E－02	1.416465 4.155973E－04	－.4164653 .8629699	－5.205816E－02 .6784715

1.499998E－02	1.431807	－.4318073	－5.397591E－02
	3.624231E－04	.9163539	.7445361
1.599998E－02	1.433018	－.4330175	－5.412719E－02
	3.082346E－04	.9623539	.8054661

运行结束有 6 个可选项可供选择。它们是

1　继续运行（Continue Run）

2　修改初始条件（Modify Initial Conditions）

3　修改定时和输出模块号（Modify Timing and Output Block Number）

4　重新运行（Restart Run）

5　结束运行（End of Run）

6　绘制曲线（Plot）

键入 1～6 中任一数字，并回车，即转入相应程序。如键入 1，则显示

Enter New Total Time

送入新的总仿真时间后就会继续运行。

键入 2，则显示

Enter：Block，Par－1，Par－2，Par－3

送入需改参数的块号及相应参数。可同时修改多个模块的参数。结束标志为 4 个 0。

键入 3，显示

Enter：Integrate Interval，Total Time

Print Interval，OP1，OP2，OP3，OP4，OP5，OP6

依次序送入相应积分步长等参数及输出模块的号数。

键入 4，重新运行。键入 5，结束运行并退到 BASIC 提示层。

键入 6，显示

Input Block Output To Plot（1 to 6）

送入需绘制曲线的模块号后，计算机询问

Polt Time Index：From，to

输入相应时间后，即能打印曲线。

3 基于离散化模型的连续系统数字仿真

上一章介绍的各种数值积分算法都是把微分方程模型转化为不同的迭代算法。从本质上来看，它们都是把连续系统离散化的方法。本章介绍的是采用离散化模型作为仿真模型，从而直接替代连续系统数学模型的仿真方法。与采用数值积分算法求解微分方程相比较，由于离散化模型用常系数差分方程代替常系数微分方程，因此，在计算机上进行运算十分简单方便，计算工作量大大下降。本章除讨论各种离散化方法并进行比较外，还介绍了有关的仿真程序。

3.1 连续系统的离散化模型

3.1.1 采样定理

如图 3－1 所示，连续系统的离散化是指在一个连续系统的输入和输出端加入周期为 T_s 的虚拟采样开关，系统的输入 $u(t)$ 经采样开关后离散化为 $u^*(t)$，系统的输出 $y(t)$ 经采样开关后离散化为 $y^*(t)$。为使离散化的系统模型与原连续系统的模型保持一致，除了加入采样开关外，还必须在输入 $u(t)$ 离散化后，送入一个虚拟的称为保持器的装置。保持器的目的是使输入信号 $u^*(t)$ 在采样间隔内仍保持连续，从频域分析来看，就是滤去离散化产生的高频分量，保持主频成分。选择合适的采样周期 T_s 及合适的保持器，就可以使离散化后的输出 $y^*(t)$ 在各采样时刻的值与 $u(t)$ 输入到原连续系统时，系统的输出 $y(t)$ 基本一致。这种方法称为连续系统的离散化。示图如图 3－1。离散化所得的模型称离散化模型。

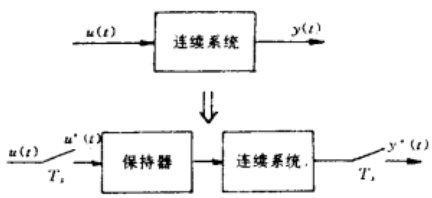


图 3－1 离散化

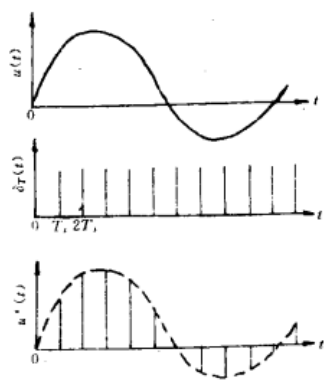


图 3－2 采样过程

香农（Shannon）采样定理给出了从采样信号恢复到原信号所必须的最低采样频率。下面分析采样频率对信号复现的影响。如图 3－2 所示。连续信号 $u(t)$，经采样后脉冲信号序列为 $u^*(t)$，则：

$$u^*(t) = u(t) \cdot \delta_T(T_s)$$

式中，$\delta_T(T_s)$ 是单位脉冲序列，具有下列特性：

①$\delta_T(T_s) = \sum_{n=-\infty}^{+\infty} \delta(t - nT_s)$

②$\int_0^{\infty} \delta(t)dt = 1$

③$\delta_T(T_s)$ 是周期函数，展开成傅氏级数为：

$$\delta_T(T_s) = \sum_{n=-\infty}^{+\infty} C_n e^{j\omega_s nt}$$

C_n 是傅氏级数系数。$\omega_s = \frac{2\pi}{T_s}$。由于

$$C_n = \frac{1}{T_s}\int_{-\frac{T_s}{2}}^{\frac{T_s}{2}} \delta_T(t) e^{-j\omega_s nt}dt$$

代入上式得：

$$\delta_T(T_s) = \frac{1}{T_s}\sum_{n=-\infty}^{+\infty} e^{j\omega_s nt}$$

$$\therefore \quad u^*(t) = \frac{1}{T_s}\sum_{n=-\infty}^{+\infty} u(t)e^{j\omega_s nt}$$

两边进行拉氏变换得：

$$u^*(s) = \frac{1}{T_s}\sum_{n=-\infty}^{+\infty} u(s + jn\omega_s)$$

其频率特性是：

$$u^*(j\omega) = \frac{1}{T_s}\sum_{n=-\infty}^{+\infty} u[j(\omega + n\omega_s)] \tag{3-1}$$

式中，$u^*(j\omega)$ 是 $u^*(t)$ 的频谱，$u(j\omega)$ 是 $u(t)$ 的频谱。可见，$u^*(t)$ 包含了 $u(t)$ 的频谱（主频）还包含了高频频谱。设 $u(t)$ 的频谱 $u(j\omega)$ 如图 3－3 所示。则得 $u^*(j\omega)$ 如图 3－4 所示。

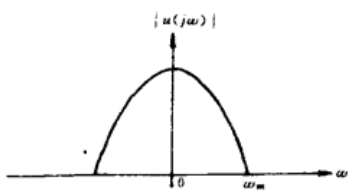


图 3－3　u（t）的频谱

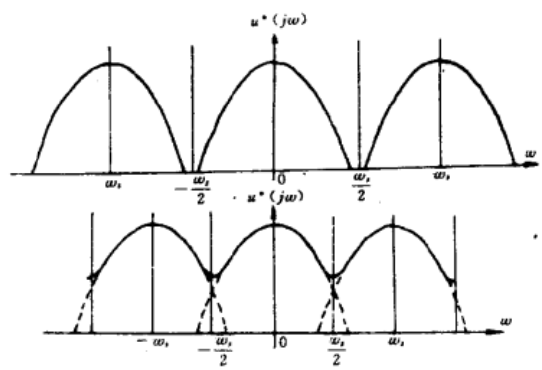


图 3－4　$u^*(t)$ 的频谱

可以看到，$u^*(t)$ 具有下列特征：

①周期函数，重复频率是 ω_s。

②当 $u(j\omega)$ 的最高频率为 ω_m 时，若 $\omega_s > 2\omega_m$，则 $u^*(j\omega)$ 高频部分不发生混叠，如图 3－4 上图所示。反之，若 $\omega_s < 2\omega_m$，则高频频谱与原信号频谱发生混叠，这时就不能忠实地复现原信号。因此，香农采样定理指出，要使采样信号恢复到原来的信号，采样频率 ω_s 必须大于 2 倍的连续信号最高频率 ω_m。

3.1.2 离散化模型

与连续系统的数学模型相类似，连续系统的离散化模型与离散时间系统的数学模型相同。常用的有差分方程、脉冲传递函数或状态差分方程等几种。

1. 差分方程

形如（3－2）式的方程称为差分方程。

$$y(k)+a_1y(k-1)+\cdots+a_ny(k-n)=b_0u(k-d)+b_1u(k-d-1)+\cdots+b_nu(k-d-n) \quad (3-2)$$

式中

$$y(k)=y(kT_s);u(k-d)=u[(k-d)T_s]$$

T_s 是采样周期。系统的纯滞后 $\tau=dT_s$。

应该指出差分方程中的系数 a_1，a_2，…，b_0，b_1，…与相对应的连续系统的系数值不同。

2. 脉冲传递函数

对（3－2）式的两边取 z 变换，可得系统的脉冲传递函数 $G(z)$。

$$G(z)=\frac{Y(z)}{U(z)}=\frac{B(z^{-1})}{A(z^{-1})}z^{-d}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_na^{-n}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}}z^{-d} \quad (3-3)$$

式中，$A(z^{-1})=1+a_1z^{-1}+\cdots+a_nz^{-n}$

$B(z^{-1})=b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_na^{-n}$

3. 状态差分方程

离散控制系统的数学模型也可以用状态差分方程的形式描述，如下式所示：

$$\begin{cases}x(k+1)=\Phi x(k)+\Gamma u(k-d)\\ y(k)=Cx(k)+Du(k-d)\end{cases} \quad (3-4)$$

或者表示为

$$\begin{cases}zx(z)=\Phi x(z)+\Gamma u(z)z^{-d}\\ y(z)=Cx(z)+Du(z)z^{-d}\end{cases} \quad (3-5)$$

选用不同的状态变量，得到的矩阵 Φ、Γ、C、D 可不同。而表征系统输入输出关系的脉冲传递函数 $G(z)$：

$$G(z)=\frac{Y(z)}{U(z)}=[C(zIn-\Phi)^{-1}\Gamma+D]z^{-d} \quad (3-6)$$

则不变。

3.2 离散化模型的特性分析

3.2.1 稳定性

在线性连续系统中，如果闭环传递函数的极点位于 s 平面的左半部分，则该系统是稳定的。对于离散或离散化的系统，若所有极点位于 z 平面的单位圆内，则该离散或离散化系统是稳定的。

如果一个系统的极点是 a_1、a_2、…，则该系统的脉冲响应的 z 变换表达式是：

$$x(z)=\frac{b_1}{z-a_1}+\frac{b_2}{z-a_2}+\cdots$$

利用留数定理可求出脉冲响应的离散时间表达式：

$$x(iT_s) = b_1a_1^{i-1} + b_2a_2^{i-2} + \cdots$$

因此，当 $|a_1|<1$、$|a_2|<1$、…，$x(iT_s)$ 才能收敛于零。这表明该系统是稳定的。若有任一极点 $|a_i|>1$，则 $x(iT_s)$ 趋于无穷，系统就是不稳定的。

判别离散或离散化系统的稳定性有很多判据，常用的有劳斯—古尔维茨判据。它是通过双线性变换：

$$z = \frac{w+1}{w-1}$$

把 z 复平面映射到与 s 复平面相类似的 w 复平面，即 $H(z)\rightarrow H(w)$。

经双线性映射，z 复平面上的单位圆内映射成为 w 复平面上的左半平面。因此，根据 $H(w)$ 的极点在 w 复平面的位置，用连续系统的劳斯—古尔维茨稳定判据即可判别离散系统 $H(w)$ 的稳定性。

3.2.2 保持器

我们知道，在连续系统离散化时，除了加入虚拟采样开关外，还要在采样开关后加入保持器，它可以把离散后的脉冲信号转换为连续信号。应该指出，在实际应用中，也并非一定要把离散信号转换为连续信号，如采用开关控制，则只要两个或三个逻辑量输出。脉宽控制则把幅值转换成脉冲的数目。步进电机和数字同步器组合成的系统实际上是一个完整的伪离散系统。但在大多数应用场合，都是用保持器把脉冲信号转换为连续信号。

从离散的脉冲信号完全精确地重现原来的连续信号是不可能的，只能是近似地重现原信号。实际应用中，根据信号的不同形式，采用不同类型的保持器，能够减小离散化造成的误差。例如尽可能减少离散化的环节数，即减少保持器的个数、采用高精度的保持器、采用较小的采样周期均能有效地减小离散化造成的误差。

保持器是具有一定幅频、相频特性的环节，它的加入必然会对离散化模型产生一定的影响，研究、分析保持器的这些幅频、相频特性有利于合理选用保持器。

3.2.2.1 零阶保持器

零阶保持器是按常数外推的装置，它把在 iT_s 时刻的信号值保持到下一个时刻 $(i+1)T_s$。根据零阶保持器的这一特性，它的传递函数可求得。设零阶保持器输入 $u^*(0)$、$u^*(T_s)$、…，输出为 $y(t)$。则有：

$$y(t) = u^*(0)[1(t) - 1(t-T_s)] + u^*(T_s)[1(t-T_s) - 1(t-zT_s)] + \cdots$$

或写成

$$y(s) = \frac{1-e^{-sT_s}}{s}u^*(s)$$

因此，零阶保持器的传递函数 $H(s)$

$$H(s) = \frac{1-e^{-sT_s}}{s} \tag{3-7}$$

它的频率特性可由 $s=j\omega$ 代入得到。

$$H(j\omega) = \frac{1-e^{-j\omega T_s}}{j\omega} = T_s\frac{\sin\left(\frac{\omega T_s}{2}\right)}{\frac{\omega T_s}{2}}e^{-j\omega\frac{T_s}{2}} \tag{3-8}$$

图 3－5 是零阶保持器幅频、相频特性。可以看到零阶保持器是一个低通滤波器，它以 $\omega_s = \frac{\pi}{\frac{T_s}{2}}$ 为周期，幅值逐渐衰减。它约产生 $\frac{T_s}{2}$ 的纯滞后，因此，离散化后的系统稳定性低于原来的连续系统。

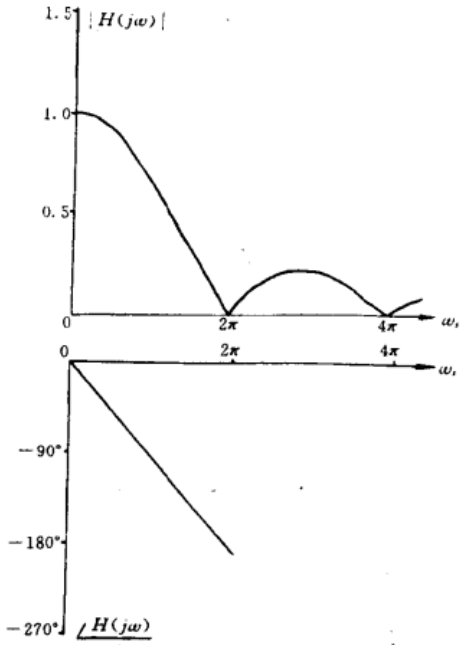


图 3－5　零阶保持器频率特性

3.2.2.2　一阶保持器

一阶保持器不是按常数外推，而是根据两个采样值作一阶差分规律的外推，即：

$$y\left[iT_s+t\right]=y\left[iT_s\right]+a\cdot\frac{y\left[iT_s\right]-y\left[(i-1)T_s\right]}{T_s}\cdot t \quad 0\leqslant t\leqslant T_s$$

上式表明，当 $a=0$ 时为零阶保持器，当 $a=1$ 时为一阶保持器。当 $t=T_s$ 时，可得：

$$y\left[(i+1)T_s\right]=y\left[iT_s\right]+a\left[y(iT_s)\right]-y\left[(i-1)T_s\right] \qquad (3-9)$$

这表明，在 $(i+1)T_s$ 的值不仅与 iT_s 时刻的值有关，而且与 $(i-1)T_s$ 时刻的值有关。即每一采样时刻的值对其后二拍都有作用。由式（3－9）可知，$y(iT_s)$ 对其后一拍的影响是 $y(iT_s)+ay(iT_s)$，对其后二拍的影响是 $-ay(iT_s)$。据此可得一阶保持器在输入为单位脉冲信号时的输出响应，如图 3－6 所示。

求得一阶保持器的输出表达式为：

$$y(t)=1(t)+\frac{a}{T_s}t-(1+a)1(t-T_s)-\frac{2a}{T_s}(t-T_s)1(t-T_s)+a1(t-2T_s)+\frac{a}{T_s}(t-2T_s)1(t-2T_s)$$

一阶保持器传递函数是（$a=1$）：

$$H(s)=\frac{1}{s}+\frac{a}{T_s s^2}-\frac{1+a}{s}e^{-sT_s}-\frac{2a}{T_s s^2}e^{-sT_s}+\frac{a}{s}e^{-2sT_s}+\frac{a}{T_s s^2}e^{-2sT_s}$$

$$=\frac{1}{s}(1-e^{-sT_s})\left[1-ae^{-sT_s}+\frac{a}{sT_s}(1-e^{-sT_s})\right]$$

$$=T_s(1+T_s s)\left(\frac{1-e^{-sT_s}}{sT_s}\right)^2 \qquad (3-10)$$

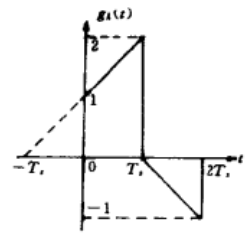


图 3－6　一阶保持器响应

其频率特性是：

$$H(j\omega)=T_s\frac{\sin(\omega T_s/2)}{\omega T_s/2}\left[1-a+a(1+j\omega T_s)\frac{\sin\frac{\omega T_s}{2}}{\frac{\omega T_s}{2}}e^{-j\omega\frac{T_s}{2}}\right]e^{-j\omega\frac{T_s}{2}} \qquad (3-11)$$

$$=T_s(1+j\omega T_s)\left[\frac{\sin\frac{\omega T_s}{2}}{\frac{\omega T_s}{2}}e^{-j\omega\frac{T_s}{2}}\right]^2$$

一阶保持器的频率特性如图 3－7 所示。可以看到，一阶保持器也是一种低通滤波器，由于它采用前两个采样值的一阶差分外推，所以有较大的失真，除对斜坡输入信号，可无失真地重现外，对一般连续信号均比零阶保持器的复现性要差，因此，实际应用中一般不采用。

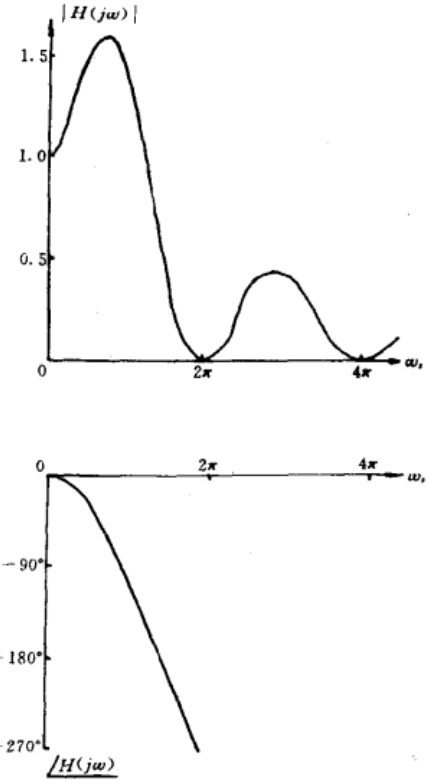


图 3－7　一阶保持器频率特性

3.2.2.3　三角形保持器

三角形保持器又称二点线性连接器，它实际上是一种内推器。在单位阶跃输入时，保持器输出 $y(t)$ 是：

$$y(iT_s + t) = y(iT_s) + \frac{y[(i+1)T_s] - y[iT_s]}{T_s} \cdot t \qquad 0 \leqslant t \leqslant T_s$$

由于这种保持器要求在 iT_s 时刻计算 iT_s 到 $(i+1)T_s$ 间的输出 $y(iT_s + t)$ 时，要已知 $y[(i+1)T_s]$ 的值，而通常这是不可能的，所以实际应用的是滞后一拍的三角形保持器。它的输出由：

$$y(iT_s + t) = y[(i-1)T_s] + \frac{y[iT_s] - y[(i-1)T_s]}{T_s} \cdot t \qquad 0 \leqslant t \leqslant T_s$$

确定。三角形保持器传递函数是

$$H(s) = \frac{(1 - e^{-sT_s})^2}{T_s s^2} \qquad (3-12)$$

它的频率特性是：

$$H(j\omega) = \left[\frac{\sin\frac{\omega T_s}{2}}{\frac{\omega T_s}{2}} e^{-j\omega\frac{T_s}{2}}\right]^2 \cdot \frac{1}{T_s}$$

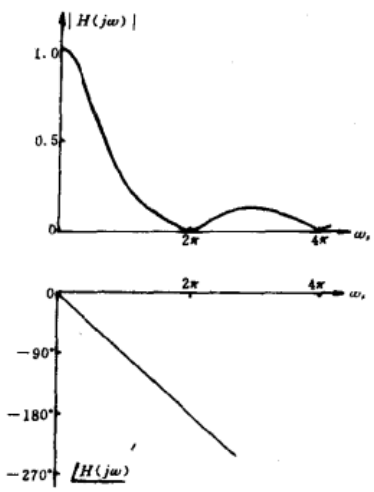


图 3－8　滞后一拍三角形保持器频率特性

三角形保持器脉冲传递函数的图形是一个三角形，因此被称为三角形保持器。滞后一拍的三角形保持器频率特性如图 3－8 所示。

可以看到，经滞后一拍的修正，三角形保持器引入了相位滞后，但它的高频部分幅值较小，比零阶或一阶保持器的幅频特性要好。

由于实际使用的保持器与理想滤波器有差别，因此，不可能不失真地复现原信号。为使信号失真减小，常可选用较高的采样频率，选用合适的保持器。

3.3 离散化的方法

连续系统有很多种离散化的方法。表征连续系统的特性可有很多参数，如增益、零、极点、稳定裕度、带宽等。在连续系统离散化时，不是所有这些参数都能得到保证，在选用离散化的方法时，应考虑这些影响。

3.3.1 z 变换法

3.3.1.1 连续系统的脉冲传递函数

设连续系统为 $G(s)$，系统的输入序列为 $u^*(iT_s)$，利用卷积公式可以求出在离散时刻连续系统的响应为：

$$y(iT_s) = \sum_{k=0}^{i} G(kT_s)u^*(i-k)T_s$$

令

$$y^*(s) = \sum_{i=0}^{\infty} y(iT_s)e^{-isT_s}$$

可以证明：

$$y^*(s) = G^*(s)u^*(s)$$

式中，$G^*(s) = \sum_{i=0}^{\infty} G(iT_s)e^{-isT_s}$

用 $z = e^{sT_s}$ 代入可得：

$$y(z) = G(z)u^*(z) \tag{3-13}$$

上式用输出信号的 z 变换表达式描述了连续系统在离散点上的特性。若连续系统是由简单的环节表示时，可以直接通过查表求得相应的 z 变换表达式。对于复杂系统可采用部分分式展开，然后查表来求得。

连续系统离散化的 z 变换法可分两类，一类是连续系统直接进行 z 变换求取脉冲传递函数。一类则对连续系统先加入保持器，然后对合成后的系统进行 z 变换。它们都是原系统的一种近似。

3.3.1.2 z 变换

这种离散化方法是直接对连续系统进行 z 变换。设连续系统用

$$G(s) = \frac{y(s)}{u(s)} = \frac{A_1}{s+a_1} + \frac{A_2}{s+a_2} + \cdots$$

表示。根据 z 变换表可直接求出形如 $\frac{A}{s+a}$ 的系统的脉冲传递函数是 $\frac{A}{1-e^{-aT_s}z^{-1}}$。因此，连续系统的脉冲传递函数可直接求得：

$$G(z) = \frac{y(z)}{u(z)} = \frac{A_1}{1-e^{-a_1T_s}z^{-1}} + \frac{A_2}{1-e^{-a_2T_s}z^{-1}} + \cdots \tag{3-14}$$

可以看到，采用 z 变换法可使 $G(z)$ 与 $G(s)$ 有相同的脉冲响应。若 $G(s)$ 是稳定的，则 $G(z)$ 也是稳定的。该方法的缺点是由于 $G(z)$ 把 ω_s 的整数倍频率变换到 z 平面上的相同频率，因此，$G(z)$ 不能保持 $G(s)$ 的频率响应，并因此发生混迭。减少混迭可以采用增大采样频率或者加入滤波器。

3.3.1.3 z 变换加入工保持器

若连续系统离散化时，加入人工的采样开关和保持器，然后进行 z 变换，则可得相应的脉

冲传递函数。对于不同的保持器，同一个连续系统可有不同的脉冲传递函数。

1．零阶保持器

零阶保持器的传递函数是：

$$H(s) = \frac{1 - e^{-sT_s}}{s}$$

因此，连续系统 $G(s)$，采用 z 变换加零阶保持器时的脉冲传递函数可求得。

$$G(z) = Z[H(s)G(s)] = Z\left[\frac{1 - e^{-sT_s}}{s}G(s)\right] = \frac{z-1}{z}Z\left[\frac{G(s)}{s}\right] \qquad (3-15)$$

对于不同的 $G(s)$，要求出 $H(s)G(s)$，然后再经分部分式进行分解，才能查表。

2．滞后一拍三角形保持器

如上述，可求出连续系统 $G(s)$，采用滞后一拍三角保持器时的脉冲传递函数。

$$G(z) = Z[H(s)G(s)] = Z\left[\frac{(1 - e^{-sT_s})^2}{T_s s^2}G(s)\right] = \frac{(z-1)^2}{z}Z\left[\frac{G(s)}{T_s s^2}\right] \qquad (3-16)$$

采用 z 变换加人工保持器的方法，由于不能直接查表，计算工作量较大，但它可改善混迭的影响。若 $G(s)$ 是稳定的，则 $G(z)$ 也稳定。由于加入了保持器，因此，$G(z)$ 不能保持 $G(s)$ 的脉冲响应，也不能保持 $G(s)$ 的频率响应。

3.3.1.4　典型环节的脉冲传递函数

由于不加入保持器的典型环节的 z 变换可以方便地查表求出，因此，下面仅讨论加入保持器的典型环节的 z 变换。

(1)积分环节

积分环节的传递函数 $G(s) = \dfrac{K}{s}$

根据(3－15)式，求得加入零阶保持器时的脉冲传递函数：

$$G(z) = \frac{z-1}{z}Z\left[\frac{G(s)}{s}\right] = \frac{z-1}{z}Z\left[\frac{K}{s^2}\right] = \frac{(z-1)}{z} \cdot \frac{KT_s z^{-1}}{(1 - z^{-1})^2} = \frac{T_s K}{z-1}$$

即
$$\frac{Y(z)}{U(z)} = \frac{T_s K}{z-1}$$

或
$$zY(z) - Y(z) = T_s K U(z)$$

写成差分方程形式是：

$$y(i+1) = y(i) + T_s K u(i) \qquad (3-17)$$

根据(3－16)式，求得加入滞后一拍三角形保持器时的脉冲传递函数：

$$G(z) = \frac{(z-1)^2}{z}Z\left[\frac{K}{T_s s^3}\right] = \frac{1}{z-1}\left[\frac{KT_s}{2} + \frac{KT_s}{2}z^{-1}\right]$$

或
$$zy(z) - y(z) = \frac{KT_s}{2}u(z) + \frac{KT_s}{2}z^{-1}u(z)$$

差分方程形如：

$$y(i+1) = y(i) + \frac{KT_s}{2}u(i) + \frac{KT_s}{2}u(i-1) \qquad (3-18)$$

(2)惯性环节

惯性环节的传递函数 $G(s) = \dfrac{K}{Ts+1}$

根据(3－15)式，得加入零阶保持器时脉冲传递函数是：

$$G(z) = \frac{z-1}{z}Z\left[\frac{K}{(Ts+1)s}\right] = \frac{K(1-e^{-T_s/T})}{Z-e^{-T_s/T}}$$

差分方程形式是：

$$y(i+1) = e^{-\frac{T_s}{T}}y(i) + k(1-e^{-\frac{T_s}{T}})u(i) \tag{3-19}$$

根据(3－16)式，得加入滞后一拍三角形保持器时的脉冲传递函数：

$$G(z) = \frac{(z-1)^2}{z}Z\left[\frac{K}{(Ts+1)T_s s^2}\right]$$

$$= \frac{[KT_s - KT(1-e^{-T_s/T})]z}{(z-e^{-Ts/T})T_s} + \frac{KT(1-e^{-T_s/T}) - KT_s e^{-T_s/T}}{(z-e^{-T_s/T})T_s}$$

差分方程形式是：

$$y(i+1) = e^{-\frac{T_s}{T}}y(i) + \left[K - \frac{KT}{T_s}(1-e^{-\frac{T_s}{T}})\right]u(i+1)$$

$$+ \left[\frac{KT}{T_s}(1-e^{-\frac{T_s}{T}}) - ke^{-\frac{T_s}{T}}\right]u(i) \tag{3-20}$$

3.3.1.5　高阶环节的脉冲传递函数

对于高阶环节，直接求取加入保持器时的脉冲传递函数，可采用部分分式分解成典型环节，然后求解。但在仿真应用中也常采用二次近似的方法，如图 3－9 所示。如采用零阶保持器对原系统 $G(s)$ 作一次近似，则有 $G(z) = Z\left[\frac{G(s)}{s}\right]\cdot\frac{z}{z-1}$。所谓二次近似则是对 z 变换式中的 $G(s)$ 与 $\frac{1}{s}$（对本例）分别再离散化，求其离散模型。如二次近似用零阶保持器，则有：

$$Z\left[\frac{G(s)}{s}\right] \approx Z[G(s)]\cdot Z\left[\frac{1-e^{-sT_s}}{s}\cdot\frac{1}{s}\right]$$

相应地，有：

$$G(z) = \frac{z}{z-1}\cdot Z\left[\frac{G(s)}{s}\right] = \frac{z}{z-1}Z\left[\frac{1-e^{-sT_s}}{s}\cdot\frac{1}{s}\right]Z[G(s)]$$

$$= \frac{T_s}{z}Z[G(s)]$$

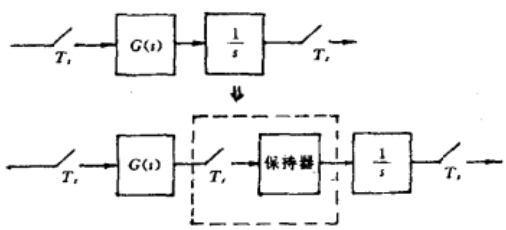


图 3－9　零阶保持器作二次近似

下面讨论二次近似用零阶保持器的情况。若一次近似加入零阶保持器，则一次近似后得

$$G(z) = \frac{z-1}{z}Z\left[\frac{G(s)}{s}\right]$$

二次近似用零阶保持器，对象为 $\frac{1}{s}$，所以有：

$$G(z) = \frac{z-1}{z}Z\left[\frac{G(s)}{s}\right] = \frac{z-1}{z}Z\left[\frac{1-e^{-sT_s}}{s}\cdot\frac{1}{s}\right]\cdot Z[G(s)]$$

$$= \frac{z-1}{z}\cdot\frac{T_s}{z-1}Z[G(s)] = \frac{T_s}{z}Z[G(z)]$$

①积分环节：$G(s) = \frac{K}{s}$

其一次近似为：$G(z)=\frac{z-1}{z}Z\left[\frac{K}{s^2}\right]=\frac{T_sK}{z-1}$

二次近似为：$G(z)=\frac{T_s}{z}Z\left[\frac{K}{s}\right]=\frac{T_s}{z}\cdot\frac{Kz}{z-1}=\frac{T_sK}{z-1}$

可见，二者结果是一致的。

②惯性环节：$G(s)=\frac{K}{Ts+1}$

其一次近似为：$G(z)=\frac{K(1-e^{-T_s/T})}{z-e^{-T_s/T}}$

二次近似为：$G(z)=\frac{T_s}{z}Z\left[\frac{K}{Ts+1}\right]=\frac{T_s\ kz}{z(z-e^{-T_s/T})T}=\frac{kT_s/T}{z-e^{-T_s/T}}$

当一次近似式中的 $e^{-\frac{T}{T}}$ 用泰勒级数前二项近似，即：$e^{-\frac{T}{T}}\approx 1+\frac{T_s}{T}$，则代入可得二次近似式。

若一次近似加入滞后一拍的三角形保持器，则一次近似后得：

$$G(z)=\frac{(z-1)^2}{z}Z\left[\frac{G(s)}{T_ss^2}\right]$$

二次近似加入零阶保持器，则

$$\begin{aligned}Z\left[\frac{G(s)}{T_ss^2}\right]&=Z\left[\frac{1-e^{-sT_s}}{s}\cdot\frac{1}{T_ss^2}\right]\cdot Z[G(s)]\\&=\left[\frac{T_s^2z(z+1)}{2T_s(z-1)^3}-\frac{T_s^2(z+1)}{2T_s(z-1)^3}\right]\cdot Z[G(s)]\\&=\frac{T_s(z+1)}{2(z-1)^2}Z[G(s)]\end{aligned}$$

代入得：

$$G(z)=\frac{T_s(z+1)}{2z}Z[G(s)] \qquad (3-21)$$

①积分环节：$G(s)=\frac{K}{s}$

一次近似式：$G(z)=\frac{KT_s(z+1)}{2z(z-1)}$

二次近似式：$G(z)=\frac{T_s(z+1)}{2z}\cdot Z\left[\frac{K}{s}\right]$

$$=\frac{T_sK(z+1)}{2(z-1)}$$

可见，二者之间相差一拍。

②惯性环节：$G(s)=\frac{K}{Ts+1}$

一次近似式：$G(z)=\frac{[KT_s-KT(1-e^{-T_s/T})]z}{(z-e^{-T_s/T})T_s}+\frac{KT(1-e^{-T_s/T})-KTe^{-T_s/T}}{(z-e^{-T_s/T})T_s}$

二次近似式：$G(z)=\frac{T_s(z+1)}{2z}\cdot Z\left[\frac{K}{Ts+1}\right]$

$$=\frac{T_sK(z+1)}{2T(z-e^{-T_s/T})}$$

写成差分方程表达式：

$$y(i+1) = e^{-\frac{T_s}{T}}y(i) + \frac{KT_s}{2T}u(i) + \frac{KT_s}{2T}u(i+1)$$

可见，近似式取了 $u(i+1)$ 与 $u(i)$ 两者的平均值。

3.3.2 替换法

3.3.2.1 替换法的基本思想

离散化的方法是把连续系统转换为离散系统的钮带。替换法根据 $z = e^{sT_s}$ 的定义，找出一些由 s 求出 z 的方法，从而由 $G(s)$ 求出 $G(z)$。因此，替换法的基本思想是：

①对于传递函数 $G(s)$，根据 z 变换的定义，用 $z = e^{sT_s}$，代入 $G(s)$ 进行运算，可以求得 $G(z)$。

②由于 $z = e^{sT_s}$ 的关系式不利于运算，所以在一定精度条件下可以用近似关系式 $z = f(s)$ 来代替 $z = e^{sT_s}$，并由近似式代入 $G(s)$，求得近似的 $G(z)$。

③在保证一定精度要求时，可采用较大的采样周期 T_s，使替换运算及其求解时的迭代运算尽可能简便和省时。

3.3.2.2 双线性变换

双线性变换是一种有效的替换法。它又称梯形积分法。根据 z 变换定义

$$z = e^{sT_s} = e^{\frac{T_s}{2}s} / e^{-\frac{T_s}{2}s}$$

把 $e^{\frac{T_s}{2}s}$ 与 $e^{-\frac{T_s}{2}s}$ 展开成泰勒级数，取前两项，

$$e^{\frac{T_s}{2}} \approx 1 + \frac{T_s}{2}s$$

$$e^{-\frac{T_s}{2}s} \approx 1 - \frac{T_s}{2}s$$

则有：

$$z \approx \frac{1 + \frac{T_s}{2}s}{1 - \frac{T_s}{2}s}$$

$$z - \frac{T_s}{2}zs \approx 1 + \frac{T_s}{2}s$$

$$z - 1 \approx \frac{T_s}{2}s(z+1)$$

$$\therefore \quad s \approx \frac{2}{T_s} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} = \frac{2}{T_s} \cdot \frac{z-1}{z+1} \tag{3-22}$$

通过

$$s = \frac{2}{T_s} \cdot \frac{z-1}{z+1}$$

的双线性变换，把 s 复平面映射到 z 复平面。设 $s = j\beta$，则

$$z = \frac{s + \frac{2}{T_s}}{-s + \frac{2}{T_s}} = \frac{\frac{2}{T_s} + j\beta}{\frac{2}{T_s} - j\beta}$$

$$\beta \neq 0, z = 1; \beta = \infty, z = -1$$

以 $s = \alpha + j\beta$ 代入得

$$z = \frac{\frac{2}{T_s} + \alpha + j\beta}{\frac{2}{T_s} - \alpha - j\beta}$$

$$|z| = \frac{\sqrt{\left(\frac{2}{T_s} + \alpha\right)^2 + \beta^2}}{\sqrt{\left(\frac{2}{T_s} - \alpha\right)^2 + \beta^2}}$$

当 $\alpha < 0$，即在 s 复平面的左半平面，则有 $|z| < 1$，即映射到 z 复平面的单位圆内。这表明采用双线性变换时，不发生频率的混迭现象。此外，当 $G(s)$ 稳定时，经双线性变换得到的 $G(z)$ 也是稳定的。双线性变换的缺点是在连续系统用双线性变换离散化时存在非线性的频率翘曲。

设连续系统 $G(s)$ 在 s 复平面的临界频率为 ω_n，相应离散化系统 $G(z)$ 在 z 平面上的频率为 ω_z，则

$$\because \quad s = j\omega_n; \; z = e^{j\omega_z T_s}$$

$$\therefore \quad s = \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}} = \frac{2}{T_s} \cdot \frac{1 - e^{-j\omega_z T_s}}{1 + e^{-j\omega_z T_s}}$$

$$= \frac{2}{T_s}\left(\frac{e^{j\frac{\omega_z}{2}T_s} - e^{-j\frac{\omega_z}{2}T_s} \cdot e^{-j\frac{\omega_z}{2}T_s} \cdot e^{j\frac{\omega_z}{2}T_s}}{e^{j\frac{\omega_z}{2}T_s}} \Big/ \frac{e^{j\frac{\omega_z}{2}T_s} + e^{-j\frac{\omega_z}{2}T_s} \cdot e^{-j\frac{\omega_z}{2}T_s} \cdot e^{j\frac{\omega_z}{2}T_s}}{e^{j\frac{\omega_z}{2}T_s}}\right) = \frac{2}{T_s}\left(\frac{e^{j\frac{\omega_z}{2}T_s} - e^{-j\frac{\omega_z}{2}T_s}}{e^{j\frac{\omega_z}{2}T_s} + e^{-j\frac{\omega_z}{2}T_s}}\right)$$

$$= \frac{2}{T_s} j \operatorname{tg} \frac{\omega_z T_s}{2}$$

即有 $$\omega_n = \frac{2}{T_s} \operatorname{tg} \frac{\omega_2 T_s}{2} \tag{3-23}$$

上式表明，双线性变换引入了频率的非线性对应关系。解决的方法是进行预修正，然后再作双线性变换。使在所需的频率响应不发生畸变。

3.3.3 根匹配法

根匹配法是一种替换法。又称极点、零点对应法。设连续系统 $G(s)$ 可表示为：

$$G(s) = \frac{K_s(s - r_1)(s - r_2)\cdots(s - r_m)}{(s - p_1)(s - p_2)\cdots(s - p_n)}$$

即连续系统的特性可由它的增益及零、极点决定。若通过 $z = e^{sT_s}$ 的映射，把原系统的零、极点映射到 z 复平面上的零、极点，则由于它们之间的一一对应关系，可以知道，若连续系统是稳定的，则离散化的系统也是稳定的。根据系统在相同输入条件下有相同的输出特性，或者相同的频率响应，可以确定离散化系统的增益。

若离散化系统可用 $G(z)$

$$G(z) = \frac{K_z(z - z_{r_1})(z - z_{r_2})\cdots(z - z_{r_m})(z + \delta_1)\cdots(z + \delta_{n-m})}{(z - z_{p_1})(z - z_{p_2})\cdots(z - z_{p_n})} \tag{3-24}$$

表示。其中 z_{r_1}、z_{r_2}、…、z_{r_m} 及 z_{p_1}、…、z_{p_n} 对应于原系统的零、极点 r_1、r_2、…、r_m 及 p_1、p_2、…、p_n。原系统还有 $n-m$ 个无穷远零点。根据这些零点映射的情况，可分三类情况讨论。

①$n-m$ 个无穷远零点映射到 z 复平面的 $z=0$ 原点处。

其依据是无穷远处的零点 $s=r=-\infty$，经 $z=e^{sT_s}$ 映射，得 $z=e^{rT_s}=0$，即映射到原点。

对于原系统的零点及极点的映射，根据 $z=e^{sT_s}$ 可直接求得：

$$z_{r_1}=e^{r_1T_s};z_{r_2}=e^{r_2T_s};\cdots;z_{r_m}=e^{r_mT_s};$$

$$z_{p_1}=e^{p_1T_s};z_{p_2}=e^{p_2T_s};\cdots;z_{p_n}=e^{p_nT_s};$$

代入可得：

$$G(z)=\frac{K_z(z-e^{r_1T_s})(z-e^{r_2T_s})\cdots(z-e^{r_mT_s})z^{n-m}}{(z-e^{p_1T_s})(z-e^{p_2T_s})\cdots(z-e^{p_nT_s})} \tag{3-25}$$

②$n-m$ 个无穷远零点映射到 z 复平面的 $z=-1$ 处。

其依据是无穷远处的零点 $s=r=\pm j\infty$ 经映射，在 z 复平面上是 $z=-1$，或根据双线性变换公式：

$$\frac{1}{s}=\frac{T_s}{2}\frac{z+1}{z-1}$$

当 s 为无穷远处零点，有 $z=-1$。同样可得：

$$G(z)=\frac{K_z(z-e^{r_1T_s})(z-e^{r_2T_s})\cdots(z-e^{r_mT_s})(z+1)^{n-m}}{(z-e^{p_1T_s})(z-e^{p_2T_s})\cdots(z-e^{p_nT_s})} \tag{3-26}$$

③$n-m$ 个无穷远零点映射到 z 复平面的 0，与 -1 间的 $-\delta_1$、$-\delta_2$、…、$-\delta_{n-m}$ 处。

其依据是 s 复平面负实轴无穷远处的零点映射到 $z=0$，s 复平面虚轴上无穷远处的零点映射到 $z=-1$，则 s 左半平面不在实轴和虚轴上的无穷远零点映射到 z 复平面的 0 与 -1 之间。δ 值可根据对应的频率响应应相同来确定。

经验表明，无穷远处零点映射到 $z=0$ 的系统具有较大的相位超前，无穷远处零点映射到 $z=-1$ 的系统具有较大的相位滞后，而映射到 0 与 -1 间的系统有较满意的相位特性。

根匹配法中 K_z 的确定是根据系统外部特性的一致性。可分两种情况：

(1) 根据终值定理确定 K_z

对原系统及离散化系统输入相同的信号，常用的为单位阶跃信号、斜坡信号等，根据终值相等，确定 K_z。

当输入信号是单位阶跃信号时，有

$$\lim_{s\to0}s\cdot\frac{1}{s}\cdot G(s)=\lim_{z\to1}\frac{z-1}{z}\cdot\frac{z}{z-1}\cdot G(z)$$

$$\text{或}\qquad \lim_{s\to0}G(s)=\lim_{z\to1}G(z) \tag{3-27}$$

当输入信号是斜坡信号时，有

$$\lim_{s\to0}s\cdot\frac{1}{s^2}\cdot G(s)=\lim_{z\to1}\frac{z-1}{z}\cdot\frac{T_sz}{(z-1)^2}\cdot G(z)$$

$$\text{或}\qquad \lim_{s\to0}\frac{G(s)}{s}=\lim_{z\to1}\frac{T_sG(z)}{(z-1)} \tag{3-28}$$

当求得原系统的终值为零时，应改用其他类型的输入信号，使其终值不为零才能确定 K_z。例如，原系统有一个位于原点的零点，用单位阶跃信号输入时，根据终值定理得终值

为零，如改为斜坡信号，则其终值可不为零。

〔例3-1〕已知连续系统 $G(s)=\dfrac{1}{s+1}$，取 $T_s=1$。

①一个无穷远零点映射到原点 $z=0$ 处：

系统零点 $s=-1$ 映射为 $z=e^{sT_s}=e^{-1}$。

$$\therefore \quad G(z)=\frac{K_z z}{(z-e^{-1})}$$

由终值定理，取单位阶跃信号输入，得

$$\lim_{s\to 0}G(s)=\lim_{z\to 1}G(z)$$

$$\text{或} \quad \lim_{s\to 0}\frac{1}{s+1}=\lim_{z\to 1}\frac{K_z z}{z-e^{-1}}$$

$$\frac{K_z}{1-e^{-1}}=1$$

$$\therefore \quad K_z=1-e^{-1}$$

$$\therefore \quad G(z)=\frac{(1-e^{-1})z}{z-e^{-1}}$$

$$\text{或} \quad y(i)=e^{-1}y(i-1)+(1-e^{-1})u(i)$$

②无穷远零点映射到 $z=-1$ 处：

$$\therefore \quad G(z)=\frac{K_z(z+1)}{z-e^{-1}}$$

$$\text{同理有：}\lim_{z\to 1}G(z)=\frac{2K_z}{1-e^{-1}}=1$$

$$\therefore \quad K_z=\frac{1}{2}(1-e^{-1})$$

$$\therefore \quad G(z)=\frac{(1-e^{-1})(z+1)}{2(z-e^{-1})}$$

或：$y(i)=e^{-1}y(i-1)+\dfrac{1}{2}(1-e^{-1})[u(i)+u(i-1)]$

〔例3-2〕已知连续系统 $G(s)=\dfrac{s}{(s+1)^2}$，取 $T_s=1$。

取无穷远零点映射到 $z=0$。若用单位阶跃信号，因终值为零无法确定 K_z，改为斜坡信号，

$$\lim_{s\to 0}\frac{G(s)}{s}=\lim_{z\to 1}\frac{T_s G(z)}{z-1}$$

原系统零点 $s=0$ 映射到 $z=1$。原系统极点 $s=-1$ 映射到 $z=e^{-1}$，因此有：

$$G(z)=\frac{K_z(z-1)z}{(z-e^{-1})^2}$$

$$\because \quad \lim_{s\to 0}\frac{G(s)}{s}=1$$

$$\therefore \quad \lim_{z\to 1}\frac{T_s G(z)}{z-1}=\frac{K_z}{(1-e^{-1})^2}=1$$

$$\therefore \quad K_z=(1-e^{-1})^2$$

$$\therefore \quad G(z) = \frac{(1-e^{-1})^2(z-1)z}{(z-e^{-1})^2} = \frac{(1-e^{-1})^2(1-z^{-1})}{1-2e^{-1}z^{-1}+e^{-2}z^{-2}}$$

或：$y(i) = 2e^{-1}y(i-1) - e^{-2}y(i-2) + (1-e^{-1})^2[u(i)-u(i-1)]$

（2）根据频率特性确定 K_z

对所需频率 ω，原系统的幅值为 $|G(j\omega)|$，相频为 $\underline{|G(j\omega)}$，映射后的 z 复平面中，频率响应应与原系统一致。

〔例 3－3〕已知连续系统 $G(s) = \frac{s}{(s+1)^2}$，取 $T_s = 1$。

设无穷远零点映射到 $z = -\delta$ 处，$0 < \delta < 1$。则有

$$G(z) = \frac{K_z(z-1)(z+\delta)}{(z-e^{-1})^2}$$

设所需频率 $\omega = 1$，原系统频率特性是：

$$G(j\omega) = \frac{j\omega}{(j\omega+1)^2}$$

当 $\omega = 1$，有 $|G(j\omega)| = \frac{1}{2}$；$\underline{|G(j\omega)} = 0$

$z = e^{sT_s} = e^{j\omega T_s}$

当 $\omega = 1$，$T_s = 1$ 有 $z = e^j$

$$\therefore \quad |G(z)| = \left|\frac{K_z(e^j-1)(e^j+\delta)}{(e^j-e^{-1})^2}\right| = \frac{1}{2}$$

$$\underline{|G(z)} = \underline{|e^j-1} + \underline{|e^j+\delta} - 2\underline{|e^j-e^{-1}} = 0$$

解得：$\delta = 0.5272; K_z = 0.2828$

$$\therefore \quad G(z) = \frac{0.2828(z-1)(z+0.5272)}{(z-e^{-1})^2}$$

或：

$$y(i) = 2e^{-1}y(i-1) - e^{-2}y(i-2) + 0.2828u(i) - 0.1337u(i-1) - 0.1491u(i-2)$$

表 3－1 列出了根匹配法的频率特性，可见，用 0 与－1 间的映射，有较好的相频及幅频特性。

表 3－1　根匹配法的比较

ω (rad/s)		0.1	0.3	0.6	0.8	1.0	1.2	2.0
幅频特性	连续系统	0.09901	0.2752	0.4412	0.4878	0.5	0.4918	0.4
	零点配在原点	0.09537	0.2659	0.4307	0.4812	0.5	0.5001	0.449
	零点配在－1 点	0.1085	0.2996	0.4688	0.505	0.5	0.47	0.2765
	零点配在（－1，0）间	0.107	0.2955	0.464	0.501	0.5	0.474	0.3024
相频特性	连续系统	78.58	56.6	28.07	12.68	0	－10.39	－36.87
	零点配在原点	80.5	62.37	39.6	27.99	19.10	12.45	0.35
	零点配在－1 点	77.64	53.78	22.11	5.08	－6.02	－6.55	－11.17
	零点配在（－1，0）间	78.53	56.45	27.88	12.53	0.05	－9.96	－31.20

根匹配法的优点是对于稳定的连续系统，用根匹配法离散化的系统也是稳定的。稳定的条件与采样周期的大小无关。因此，可用于较大采样周期的快速数字仿真。

3.3.4 微分映射法

当连续系统用时域上的微分方程表示时，变量 y 的导数可用差分形式表示，如表 3－2 所示。

表 3－2 微分－差分转换表

连续系统	$\frac{dy}{dt}$	$\frac{d^2y}{dt^2}$
前向差分	$\frac{y(i+1)-y(i)}{T_s}$	$\frac{y(i+2)-2y(i+1)+y(i)}{T_s^2}$
后向差分	$\frac{y(i)-y(i-1)}{T_s}$	$\frac{y(i)-2y(i-1)+y(i-2)}{T_s^2}$
中心差分	$\frac{y(i+1)-y(i-1)}{2T_s}$	$\frac{y(i+1)-2y(i)+y(i-1)}{T_s^2}$

微分映射法是最简单的离散化方法。它把连续系统直接映射到 z 域。对高阶导数项可采用其他近似式或差分式。

(1) 前向差分

$$\because \quad \frac{dy}{dt}=\frac{y(i+1)-y(i)}{T_s}$$

或：

$$s=\frac{z-1}{T_s};z=1+sT_s \tag{3-29}$$

当 $T_s s=\alpha+j\beta$ 时，有 $z=1+\alpha+j\beta$，只有当 $\beta^2<1-(\alpha+1)^2$ 时才有 $|z|<1$。这表明，采用前向差分产生了不稳定的映射关系。

(2) 后向差分

$$\because \quad \frac{dy}{dt}=\frac{y(i)-y(i-1)}{T_s}$$

或：

$$s=\frac{1-z^{-1}}{T_s};z=\frac{1}{1-sT_s} \tag{3-30}$$

当 $T_s s=j\beta$ 时，有

$$z=\frac{1}{1-j\beta}=\frac{1}{2}\left(1+\frac{1+j\beta}{1-j\beta}\right)=\frac{1}{2}(1+e^{2j\mathrm{tg}^{-1}\beta})$$

z 的实部为

$$\mathrm{Re}[z]=\frac{1}{2}+\frac{\cos(2\mathrm{tg}^{-1}\beta)}{2}$$

z 的虚部为

$$\mathrm{Im}[z]=\frac{\sin(2\mathrm{tg}^{-1}\beta)}{2}$$

这表明，s 平面上的虚轴经后向差分映射成 z 复平面的圆，其圆心在 $\left(\frac{1}{2},0\right)$，半径为 $\frac{1}{2}$。因此 $G(s)$ 若稳定，则用后向差分可得稳定的 $G(z)$。但由于对频率响应的畸变，实际应用较少。

(3) 中心差分

$$\because \quad \frac{dy}{dt} = \frac{y(i+1) - y(i-1)}{2T_s}$$

$$\therefore \quad s = \frac{z - z^{-1}}{2T_s}; z = T_s s \pm \sqrt{T_s^2 - 1} \qquad (3-31)$$

可以看到稳定域也发生变化。必须在离散化后比较与原系统的频率特性。

〔例3-4〕已知连续系统的微分方程是：

$$\frac{d^2y}{dt^2} + 0.2\frac{dy}{dt} + y = u(t)$$

取 $T_s=1$，按中心差分离散得：

$$\frac{y(i+1) - 2y(i) + y(i-1)}{1} + 0.2\frac{y(i+1) - y(i-1)}{2} + y(i) = u(i)$$

或：

$$y(i+1) = 0.909y(i) - 0.818y(i-1) + 0.909u(i)$$

故有：

$$G(z) = \frac{Y(z)}{U(z)} = \frac{0.909z}{z^2 - 0.909z + 0.818}$$

3.3.5 状态变量法

3.3.5.1 基本思想

设连续系统 $\dot{x} = Ax + Bu$

其解析解是：

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

当 $t=kT_s$，有

$$x(kT_s) = e^{AkT_s}x(0) + \int_0^{kT_s} e^{A(kT_s-\tau)}Bu(\tau)d\tau$$

当 $t=(k+1)T_s$，有

$$x[(k+1)T_s] = e^{AkT_s}x(0)e^{AT_s} + \int_0^{(k+1)T_s} e^{A[(k+1)T_s-\tau]}Bu(\tau)d\tau$$

$$\therefore \quad x[(k+1)T_s] = e^{AT_s}x(kT_s) + \int_{kT_s}^{(k+1)T_s} e^{A[(k+1)T_s-\tau]}Bu(\tau)d\tau$$

作变量置换，$\tau = kT_s + t$，$\therefore d\tau = dt$

$t=\tau - kT_s$，对积分上、下限改变：

当 $\tau=kT_s$，$t=0$；当 $\tau=kT_s+T_s$，$t=T_s$；

$$\therefore \quad x[(k+1)T_s] = e^{AT_s}x(kT_s) + \int_0^{T_s} e^{A(T_s-t)}Bu(kT_s+t)dt \qquad (3-32)$$

这是离散化的基本公式。

①零阶保持器：

根据零阶保持器特点，得：

$$u(kT_s + t) = u(kT_s) \qquad 0 \leqslant t \leqslant T_s$$

$$\therefore \quad x[(h+1)T_s] = e^{AT_s}x(kT_s) + \int_0^{T_s} e^{A(T_s-t)}Bdt \cdot u(kT_s)$$

为简化，设

$$x[(k+1)T_s] = x(k+1), u(kT_s) = u(k)$$

则有：

$$x(k+1) = \Phi(T_s)x(k) + \Phi_m(T_s)u(k) \tag{3-33}$$

式中：$\Phi(T_s) = e^{AT_s}$ (3-34)

$$\Phi_m(T_s) = \int_0^{T_s} \Phi(T_s - t)Bdt \tag{3-35}$$

②三角形保持器：

如图3-10所示，在kT_s与$(k+1)T_s$间$u(kT_s+t)$的值由$u(k)$与$u(k+1)$连线上相应值代替。

根据相似关系可得：

$$\Delta u = \frac{u(k+1) - u(k)}{T_s} \cdot t = \dot{u}(k) \cdot t$$

$$\therefore \quad u(kT_s + t) = u(k) + \dot{u}(k)t$$

$$\therefore \quad x(k+1) = e^{AT_s}x(k) + \int_0^{T_s} e^{A(T_s-t)}B[u(k) + \dot{u}(k)t]dt$$

$$= \Phi(T_s)x(k) + \Phi_m(T_s)u(k) + \tilde{\Phi}_m(T_s)\dot{u}(k)$$

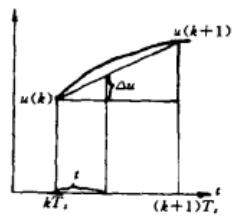


图3-10 三角形保持器的信号值

式中，$\Phi(T_s)$、$\Phi_m(T_s)$如前述，$\tilde{\Phi}_m(T_s)$是：

$$\tilde{\Phi}_m(T_s) = \int_0^{T_s} e^{A(T_s-t)}tBdt = \int_0^{T_s} \Phi(T_s - t)tBdt \tag{3-36}$$

3.3.5.2 典型环节的$\Phi(T_s)$、$\Phi_m(T_s)$、$\tilde{\Phi}_m(T_s)$

连续系统是由一些典型环节组合而成的。若对每个环节都能根据上述方法离散化，则通过逐个对环节的迭代运算就能求得数值解。下面讨论典型环节的$\Phi(T_s)$、$\Phi_m(T_s)$及$\tilde{\Phi}_m(T_s)$的求法。

①积分环节：$\frac{Y(s)}{U(s)} = \frac{K}{s}$

状态方程可写成：$\dot{x} = Ku$；输出方程：$y = x$。

$$\therefore \quad A = 0; B = K; C = 1; D = 0$$

$$\Phi(T_s) = e^{AT_s} = e^0 = 1$$

$$\Phi_m(T_s) = \int_0^{T_s} e^{A(T_s-t)}Bdt = \int_0^{T_s} Kdt = KT_s$$

$$\tilde{\Phi}_m(T_s) = \int_0^{T_s} e^{A(T_s-t)}Btdt = \int_0^{T_s} Ktdt = \frac{KT_s^2}{2}$$

所以，采用零阶保持器时有：

$$y(i+1) = y(i) + KT_s u(i) \tag{3-37}$$

采用三角形保持器时有：

$$y(i+1) = y(i) + KT_s u(i) + \frac{KT_s^2}{2}\dot{u}(i) \tag{3-38}$$

(3-37)式与z变换法的(3-17)式完全相同。而(3-38)式中，若$\dot{u}(i)$用

$$\dot{u}(i) = \frac{u(i+1) - u(i)}{T_s}$$

代入，得：

$$y(i+1) = y(i) + \frac{KT_s}{2}u(i) + \frac{KT_s}{2}u(i+1) \qquad (3-39)$$

与滞后一拍三角形保持器的结果（3－18）式比较，可以看到，（3－18）式中的输入变量确实比（3－39）式中的输入变量滞后一拍。

②惯性环节：$G(s) = \dfrac{K}{Ts+1}$

状态方程：$\dot{x} = -\dfrac{1}{T}x + \dfrac{K}{T}u$；输出方程：$y = x$

$\therefore$ $A = -\dfrac{1}{T}; B = \dfrac{K}{T}; C = 1; D = 0$。

$$\Phi(T_s) = e^{AT_s} = e^{-\frac{T_s}{T}}$$

$$\Phi_m(T_s) = \int_0^{T_s} e^{A(T_s-t)}Bdt = K(1 - e^{-\frac{T_s}{T}})$$

$$\dot{\Phi}_m(T_s) = \int_0^{T_s} e^{A(T_s-t)}Btdt = KT_s + KT(e^{-\frac{T_s}{T}} - 1)$$

所以，采用零阶保持器时有：

$$y(i+1) = e^{-\frac{T_s}{T}}y(i) + K(1 - e^{-\frac{T_s}{T}})u(i) \qquad (3-40)$$

采用三角形保持器时有：

$$y(i+1) = e^{-\frac{T_s}{T}}y(i) + K(1 - e^{-\frac{T_s}{T}})u(i) + [KT_s + KT(e^{-\frac{T_s}{T}} - 1)]\dot{u}(i) \qquad (3-41)$$

（3－40）式与（3－19）式完全相同，（3－41）式在 $\dot{u}(i)$ 用 $\dot{u}(i) = \dfrac{u(i+1) - u(i)}{T_s}$ 代入时，也与（3－20）式相同。

③比例积分环节：$G(s) = K\left(1 + \dfrac{1}{T_i s}\right)$

状态方程：$\dot{x} = \dfrac{K}{T_i}u$；输出方程：$y = x + Ku$

$\therefore$ $A=0$；$B=\dfrac{K}{T_i}$；$C=1$；$D=K$。

比照①的方法，得

$$\Phi(T_s) = 1; \Phi_m(T_s) = \frac{K}{T_i}T_s; \dot{\Phi}_m(T_s) = \frac{KT_s^2}{2T_i}$$

$$\therefore \qquad x(i+1) = x(i) + \frac{K}{T_i}T_s u(i) \qquad \text{（零阶保持器）} \qquad (3-41)$$

$$\text{或} \qquad x(i+1) = x(i) + \frac{K}{T_i}T_s u(i) + \frac{KT_s^2}{2T_i}\dot{u}(i) \qquad \text{（三角形保持器）} \qquad (3-42)$$

由于本环节中 $y \neq x$ 而是由 $y = x + Ku$ 决定，因此相应求得：$y(i+1) = x(i+1) + K \cdot u(i+1)$。

④二阶环节：$G(s) = \dfrac{A_1}{s + \dfrac{1}{T_1}} + \dfrac{A_2}{s + \dfrac{1}{T_2}} = \dfrac{(A_1 + A_2)s + \left(\dfrac{A_2}{T_1} + \dfrac{A_1}{T_2}\right)}{\left(s + \dfrac{1}{T_1}\right)\left(s + \dfrac{1}{T_2}\right)}$

状态方程：$\begin{cases} \dot{x}_1 = -\dfrac{1}{T_1}x_1 + u \\ \dot{x}_2 = -\dfrac{1}{T_2}x_2 + u \end{cases}$；输出方程：$y = A_1x_1 + A_2x_2$

$$\therefore A = \begin{bmatrix} -\dfrac{1}{T_1} & 0 \\ 0 & -\dfrac{1}{T_2} \end{bmatrix}; B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}; C = [A_1 \quad A_2]; D = 0$$

对于矩阵 A，计算指数矩阵 e^{AT_s} 有很多种方法。常用的有：

· 幂级数展开法：

$$e^{AT_s} = I + AT_s + \frac{A^2T_s^2}{2!} + \cdots + \frac{A^kT_s^k}{k!} + \cdots \tag{3-43}$$

在数字仿真中，由于用计算机计算上述级数及其和都比较方便，因此得到广泛应用。

· 拉氏变换法：

$$e^{AT_s} = \mathcal{L}^{-1}[(sI - A)^{-1}]\big|_{t=T_s} \tag{3-44}$$

求解析解时，该方法常用。

· 西尔威斯特法（Sylvester）

$$\begin{aligned} e^{AT_s} &= \sum_{i=1}^{n}\left(e^{(s_it)} \cdot \prod_{\substack{j=1 \\ j\neq i}}^{n}\frac{A - s_jI}{s_i - s_j}\right)\Bigg|_{t=T_s} \\ &= \frac{[A - s_2I]\cdots[A - s_nI]}{(s_1 - s_2)(s_1 - s_3)\cdots(s_1 - s_n)}e^{s_1t}\Big|_{t=T_s} + \frac{[A - s_1I][A - s_3I]\cdots[A - s_nI]}{(s_2 - s_1)(s_2 - s_3)\cdots(s_2 - s_n)}e^{s_2t}\Big|_{t=T_s} \\ &\quad + \cdots + \frac{[A - s_1I]\cdots[A - s_{n-1}I]}{(s_n - s_1)(s_n - s_2)\cdots(s_n - s_{n-1})}e^{s_nt}\Big|_{t=T_s} \end{aligned} \tag{3-45}$$

式中，s_i 是 A 的特征根，各不相同。若有相同重根，应采用汇合插值公式。

采用拉氏变换法求得：

$$\Phi(T_s) = e^{AT_s} = \begin{bmatrix} e^{-\frac{T_s}{T_1}} & 0 \\ 0 & e^{-\frac{T_s}{T_2}} \end{bmatrix} \tag{3-46}$$

$$\Phi_m(T_s) = \begin{bmatrix} T_1(1 - e^{-\frac{T_s}{T_1}}) \\ T_2(1 - e^{-\frac{T_s}{T_2}}) \end{bmatrix} \tag{3-47}$$

$$\hat{\Phi}_m(T_s) = \begin{bmatrix} T_1T_s + T_1^2(e^{-\frac{T_s}{T_1}} - 1) \\ T_2T_s + T_2^2(e^{-\frac{T_s}{T_2}} - 1) \end{bmatrix} \tag{3-48}$$

$$\therefore \quad \begin{cases} x_1(i+1) = e^{-\frac{T_s}{T_1}}x_1(i) + T_1(1 - e^{-\frac{T_s}{T_1}})u(i) \\ x_2(i+1) = e^{-\frac{T_s}{T_2}}x_2(i) + T_2(1 - e^{-\frac{T_s}{T_2}})u(i) \end{cases} \quad \text{（零阶保持器）}$$

或

$$\begin{cases} x_1(i+1) = e^{-\frac{T}{T_1}}x_1(i) + T_1(1-e^{-\frac{T}{T_1}})u(i) + [T_1T_s + T_1^2(e^{-\frac{T}{T_1}}-1)]\dot{u}(i) \\ x_2(i+1) = e^{-\frac{T}{T_2}}x_2(i) + T_2(1-e^{-\frac{T}{T_2}})u(i) + [T_2T_s + T_2^2(e^{-\frac{T}{T_2}}-1)]\dot{u}(i) \end{cases}$$

（三角形保持器）

而

$$y(i+1) = A_1x_1(i+1) + A_2x_2(i+1)$$

⑤超前滞后环节：$G(s) = \dfrac{K(s+b)}{(s+a)}$

状态方程：$\dot{x} = -ax + Ku$；输出方程：$y = (b-a)x + Ku$

$\therefore$ $A = -a; B = K; C = (b-a); D = K$。

$\therefore$ $$\Phi(T_s) = e^{AT_s} = e^{-aT_s} \tag{3-49}$$

$$\Phi_m(T_s) = \int_0^{T_s} e^{-a(T_s-t)} \cdot Kdt = \frac{K}{a}(1-e^{-aT_s}) \tag{3-50}$$

$$\hat{\Phi}_m(T_s) = \int_0^{T_s} e^{-a(T_s-t)} \cdot Ktdt = \frac{K}{a}\left[T_s + \frac{1}{a}(e^{-aT_s}-1)\right] \tag{3-51}$$

选用零阶保持器有：

$$x(i+1) = e^{-aT_s}x(i) + \frac{K}{a}(1-e^{-aT_s})u(i)$$

选用三角形保持器有：

$$x(i+1) = e^{-aT_s}x(i) + \frac{K}{a}(1-e^{-aT_s})u(i) + \frac{K}{a}\left[T_s + \frac{1}{a}(e^{-aT_s}-1)\right]\dot{u}(i)$$

输出

$$y(i+1) = (b-a)x(i+1) + Ku(i+1)。$$

3.3.6 增广矩阵法

3.3.6.1 基本思想

①对单输入 u 的连续系统：

$$\dot{x} = Ax + Bu$$

解析解是

$$x(t) = e^{At}x(0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$$

取

$$e^{At} = I + At + \frac{A^2t^2}{2!} + \frac{A^3t^3}{3!} + \frac{A^4t^4}{4!}$$

则解的计算精度与四阶龙格—库塔法相同。

②对已知 T_s，初值 $x(0)$ 的齐次方程 $\dot{x} = Ax$ 可直接用递推公式：

$$x(n) = \left[I + AT_s + \frac{A^2T_s^2}{2} + \frac{A^3T_s^3}{6} + \frac{A^4T_s^4}{24}\right]x(n-1) \tag{3-52}$$

而 AT_s 等各项均可事先计算好，这就大大加快了计算速度。

③对非齐次方程 $\dot{x} = Ax + Bu$，可引入状态变量 x_{n+1}、x_{n+2}、…，使 $\dot{x} = Ax + Bu$ 增广为齐次方程，从而用递推公式直接计算。

3.3.6.2 增广矩阵的实现

输入信号的形式不同，增广的状态变量也不同。

①输入信号是阶跃信号：

设阶跃信号幅值为 u_0，则有 $u(t) = u_0 \cdot 1(t)$

令状态变量 $x_{n+1}(t) = u(t) = u_0 \cdot 1(t)$，

则有，$\dot{x}_{n+1} = 0; x_{n+1}(0) = u_0$

增广矩阵成为：

$$\begin{bmatrix} \dot{x} \\ \cdots \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} A & B \\ \cdots & \cdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ x_{n+1} \end{bmatrix} \tag{3-53}$$

输出方程成为：

$$y = \begin{bmatrix} C & \vdots & 0 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ x_{n+1} \end{bmatrix} \tag{3-54}$$

$$\text{初值：} \begin{bmatrix} x(0) \\ \cdots \\ x_{n+1}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ \cdots \\ u_0 \end{bmatrix} \tag{3-55}$$

②输入信号是斜坡信号：

设斜坡信号 $u(t) = u_0 t$，斜率为 u_0。

令状态变量：$x_{n+1}(t) = u(t) = u_0 t$

$$x_{n+2}(t) = \dot{x}_{n+1}(t) = u_0$$

则有：$\dot{x}_{n+1} = u_0$；$x_{n+1}(0) = 0$；$x_{n+2}(0) = u_0$。

增广矩阵成为：

$$\begin{bmatrix} \dot{x} \\ \cdots \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \end{bmatrix} = \begin{bmatrix} A & B & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ x_{n+1} \\ x_{n+2} \end{bmatrix} \tag{3-56}$$

输出方程成为：

$$y = \begin{bmatrix} C & \vdots & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ x_{n+1} \\ x_{n+2} \end{bmatrix} \tag{3-57}$$

$$\text{初值：} \begin{bmatrix} x(0) \\ \cdots \\ x_{n+1}(0) \\ x_{n+2}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \\ u_0 \end{bmatrix} \tag{3-58}$$

③输入信号是指数信号：

设输入信号 $u(t) = u_0 e^{-t}$

令状态变量：$x_{n+1}(t) = u_0 e^{-t}$

则有：$\dot{x}_{n+1}(t) = -u_0 e^{-t} = -x_{n+1}(t); x_{n+1}(0) = u_0$

增广矩阵成为：

$$\begin{bmatrix} \dot{x} \\ \cdots \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} A & B \\ \cdots & \cdots \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x \\ \cdots \\ x_{n+1} \end{bmatrix} \tag{3-59}$$

输出方程成为：

$$y = [C \quad \vdots \quad 0]\begin{bmatrix} x \\ \cdots \\ x_{n+1} \end{bmatrix} \tag{3-60}$$

$$\text{初值：}\begin{bmatrix} x(0) \\ \cdots \\ x_{n+1}(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ u_0 \end{bmatrix} \tag{3-61}$$

〔例 3－5〕已知系统 $G(s) = \dfrac{K}{Ts+1}$

状态方程可写成：$\dot{x}_1 = -\dfrac{1}{T}x_1 + \dfrac{K}{T}u$；初值 $x_1(0) = 0$

输出方程是：$y = x_1$

设输入信号为单位阶跃信号，即 $u(t) = 1(t)$

令状态变量 $x_2(t) = u(t) = 1(t)$，

则有 $\dot{x}_2(t) = 0$；$x_2(0) = 1$

增广矩阵是：$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} -\dfrac{1}{T} & \dfrac{K}{T} \\ 0 & 0 \end{bmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

输出方程：$y = [1 \quad 0]\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$；初值：$\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

$\therefore \quad A = \begin{bmatrix} -\dfrac{1}{T} & \dfrac{K}{T} \\ 0 & 0 \end{bmatrix}$。若取 $T = 1, K = 1, T_s = 1$，则有：

$\therefore \quad A = \begin{pmatrix} -1 & +1 \\ 0 & 0 \end{pmatrix}; A^2 = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}; A^3 = \begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix}; A^4 = \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}$

$$\therefore \quad e^{AT_s} = I + AT_s + \frac{A^2T_s^2}{2} + \frac{A^3T_s^3}{6} + \frac{A^4T_s^4}{24} = \begin{pmatrix} 0.375 & 0.625 \\ 0 & 0 \end{pmatrix}$$

可得递推式：

$$\begin{pmatrix} x_1(i) \\ x_2(i) \end{pmatrix} = \begin{pmatrix} 0.375 & 0.625 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1(i-1) \\ x_2(i-1) \end{pmatrix}$$

表 3－3 列出了本法与精确解的前八步的值。可以看到，在取采样周期 $T_s = 1$ 这样大的情况下还是有很高的精度。因此，本法适于大采样周期的快速数字仿真。

表 3－3　增广矩阵法的计算值

i	0	1	2	3	4	5	6	7
本法 $x_1(i)$	0	0.625	0.859375	0.9473	0.980225	0.99258	0.99722	0.998957
精确计算值	0	0.632	0.86466	0.9502	0.98168	0.99326	0.99752	0.999088

3.4 非线性系统的数字仿真

非线性系统是指一个控制系统中包含有一个或几个有非线性特性的环节。例如，在工业生产过程中，变送器有不灵敏区的非线性特性；执行机构和控制阀有饱和非线性及滞环非线性特性。此外，有时为了控制的需要也要在控制系统中加入非线性环节，组成非线性系统。例如变增益控制；具有不灵敏区的非线性控制等。对于非线性系统，在控制原理中常用描述函数法及相平面法进行研究。考虑到采用离散化数字仿真时，是对每个环节的输入和输出进行计算的，因此，非线性环节的数字仿真可以方便地按环节的位置，安插在相应的位置进行输入和输出的运算。为此，在本节介绍非线性系统的数字仿真。

3.4.1 非线性环节的类型

非线性环节可以有各种不同的类型。其基本算式是：$y = f(u)$

式中，$f(u)$ 不是 u 的线性函数。u 是环节的输入信号，y 是环节的输出信号。应用于工业生产过程中的典型非线性环节有：

①饱和非线性；

②不灵敏区非线性；

③分段线性的非线性；

④滞环非线性；

⑤两位式继电非线性；

⑥三位式具有死区的继电非线性；

⑦具有滞环的继电非线性；

⑧变增益非线性。

图 3－11（a）－（h）分别绘制了上述 8 种典型非线性环节的输入输出特性。

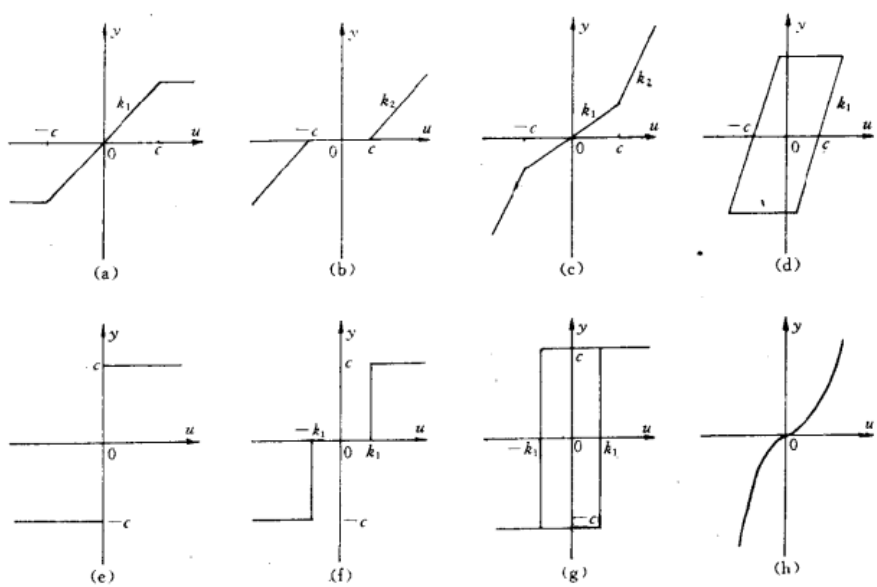


图 3－11　典型非线性环节的特性

可以看到，①②两类非线性特性是③非线性的特例。即在③非线性环节中若 $k_1 \neq 0$；$k_2 \neq 0$ 得不灵敏区非线性特性；

若 $k_1 \approx 0$；$k_2 = 0$ 得饱和非线性特性。

当三位式具有死区的继电非线性特性中的 $H = 0$，就可以得到两位式继电非线性特性。因此，下面仅讨论五类非线性环节的数字仿真程序。

3.4.2 典型非线性环节的数字仿真子程序

3.4.2.1 分段线性的非线性环节

根据图 3－11（c），可以列出分段线性的非线性环节的输入 u 与输出 y 间的关系：

$$y = \begin{cases} k_1 c + k_2(u - c) & u > c \\ k_1 u & -c \leqslant u \leqslant c \\ -[k_1 c + k_2(-u - c)] & u < -c \end{cases} \tag{3-62}$$

仿真程序框图如图 3－12 所示。

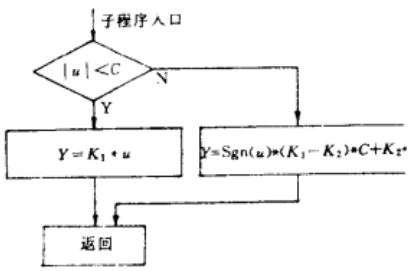


图 3－12 分段线性的非线性环节数字仿真子程序框图

考虑到程序的结构简化，可以采用符号函数 SGN（·），

$$\mathrm{SGN}(u) = \begin{cases} 1 & u > 0 \\ 0 & u = 0 \\ -1 & u < 0 \end{cases}$$

则有

$$y = \begin{cases} k_1 \cdot u & |u| \leqslant c \\ \mathrm{sgn}(u) \cdot (k_1 - k_2) \cdot c + k_2 \cdot u & |u| > c \end{cases} \tag{3-63}$$

仿真子程序成为：

```
1300  UO(I) = K1(I) * UI(I)
1310  IF ABS(UI(I)) > C(I) THEN UO(I) = SGN(UI(I)) * (K1(I) - K2(I)) * C(I) +
      K2(I) * UI(I)
1320  RETURN
```

程序中 UO（I）为输出，UI（I）是输入，I 是环节号，K1（I）、K2（I）、C（I）分别是 k_1、k_2 及 c。

3.4.2.2 滞环非线性环节

根据图 3－11（d），可以列出滞环非线性环节的输入 u 与输出 y 之间的关系。设环节本次采样值输入为 $u(k)$，计算输出 $y(k)$，上一次输入为 $u(k-1)$，输出为 $y(k-1)$。

$$y(k) = \begin{cases} K1(u(k) - c) & u(k) > u(k-1) \text{ 及 } y(k-1) \leqslant K1(u(k) - c) \\ K1(u(k) + c) & u(k) < u(k-1) \text{ 及 } y(k-1) \geqslant K1(u(k) + c) \\ y(k-1) & \text{其余} \end{cases} \tag{3-64}$$

图 3－13 是滞环非线性环节仿真子程序框图。同样采用符号函数可简化程序。其子程序如下：

```
1400  AA = SGN(UI(I,1) - UI(I,0))
1410  UO(I,1) = (UI(I,1) - AA * C(I)) * K1(I)
1420  IF UO(I,0) * AA > UO(I,1) * AA THEN UO(I,1) = UO(I,0)
1430  UO(I,0) = UO(I,1)
```

```
1440  UI(I,0)=UI(I,1)
1450  RETURN
```

程序中，UO（I，1）与UO（I，0）分别表示$y(k)$，$y(k-1)$；UI（I，1）与UI（I，0）分别表示$u(k)$、$u(k-1)$。1430、1440语句是为下次计算所用的变量替换。

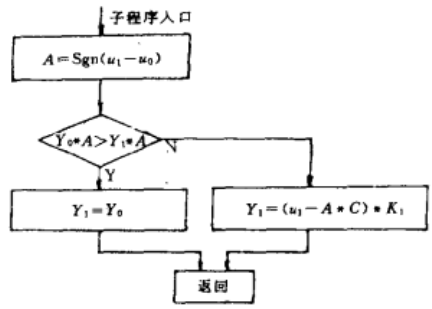


图3-13　滞环非线性环节仿真子程序框图

3.4.2.3　三位式具有死区的继电非线性环节

根据图3-11（f），可以列出三位式具有死区的继电非线性环节的输入u与输出y间的关系：

$$y=\begin{cases}c & u>k_1\\ 0 & -k_1\leqslant u\leqslant k_1\\ -c & u<-k_1\end{cases}\quad(3-65)$$

仿真程序框图如图3-14所示。采用符号函数，子程序如下：

```
1600  UO(I)=SGN(UI(I))*C(I)
1610  IF ABS(UI(I))<=K1(I) THEN UO(I)=0
1620  RETURN
```

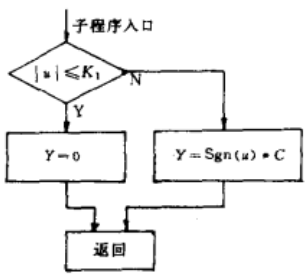


图3-14　三位式具有死区的继电非线性环节仿真子程序框图

程序中，UO（I）与UI（I）分别是环节的输出与输入。C(I)与K1(I)分别对应于c与$K1$。I是环节号。

3.4.2.4　具有滞环的继电非线性环节

根据图3-11（g），列出具有滞环的继电非线性环节的输入u与输出y间的关系。同样，设本次采样输入值为$u(k)$，计算的输出值为$y(k)$。上次采样输入值为$u(k-1)$，输出为$y(k-1)$。则有：

$$y=\begin{cases}c & u(k)>u(k-1)\text{ 及 }u(k)\geqslant K1\\ -c & u(k)<u(k-1)\text{ 及 }u(k)\leqslant -K1\\ y(k-1) & \text{其余}\end{cases}\quad(3-66)$$

程序框图见图3-15。子程序如下：

```
1700  AA=SGN(UI(I,1)-UI(I,0))
1710  UO(I,1)=AA*C(I)
1720  IF ABS(UI(I,1))<K1(I) THEN UO
(I,1)=UO(I,0)
1730  UI(I,0)=UI(I,1)
1740  UO(I,0)=UI(I,1)
1750  RETURN
```

程序中，UI(I,0)与UI(I,1)分别为$u(k-1)$及$u(k)$。UO(I,0)与UO(I,1)分别为$y(k-1)$及$y(k)$。I是环节号。C(I)与K1(I)对应于c及$K1$。

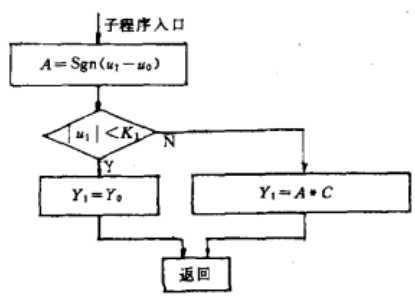


图3-15　具有滞环的继电非线性环节仿真子程序框图

3.4.2.5　变增益非线性环节

根据图 3－11（h），列出变增益非线性环节的输入 u 与输出 y 间的关系：

$$y = \begin{cases} Ku^2 & u > 0 \\ -Ku^2 & u < 0 \end{cases}$$

这类非线性环节常用于非线性环节的补偿。如在 pH 控制中作为非线性控制器的变增益环节，这时，输入是偏差信号。变增益非线性的一般形式是：

$$y = \begin{cases} K \mid u \mid^n + B & u > 0 \\ -K \mid u \mid^n - B & u < 0 \end{cases} \tag{3-67}$$

式中，B 是大于或等于 0 的实数，n 是大于 0 的实数。在采用数字控制器的控制系统中，这种算式易于实施，而得到广泛应用。当 $n=0$ 得到两位式的继电非线性特性。当 $n=1$ 时，得到输出有死区的非线性特性。

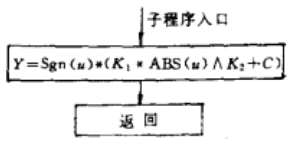


图 3－16　变增益非线性环节仿真子程序框图

变增益非线性环节仿真子程序如下：

```
1800  UO(I) = (K1(I) * ABS(UI(I)) ∧ K2(I) + C(I)) *
      SGN(UI(I))
1810  RETURN
```

程序中，K1(I)、K2(I)及 C(I)对应于 K、n 及 B。

程序框图如图 3－16 所示。

3.5　离散化模型的数字仿真程序

3.5.1　离散化模型的合成

与 2.1.4 数学模型的合成方法相同，在离散化模型的合成时，首先将系统分成若干环节，设第 I 个环节的输入信号是 UI（I），输出信号是 UO（I）。该环节输出与输入的关系由离散化后该环节的特性决定。然后，列出表示各环节间的连接关系的连接矩阵 P 与表示各环节与外部输入 M 的连接关系的连接矩阵 F：

$$UI = F \cdot M + P \cdot UO \tag{3-68}$$

$$UI = [UI(1), UI(2), \cdots, UI(n)]^T; UO = [UO(1), UO(2), \cdots, UO(n)]^T$$

式中，n 为环节数，外部输入信号个数为 nr，因此，F 是 $n \times nr$ 维矩阵，P 是 $n \times n$ 维矩阵。F_{ij} 表示第 j 个外输入信号对第 i 个环节的连接系数。P_{ij} 表示第 j 个环节对第 i 个环节的连接系数。F_{ij} 与 P_{ij} 的取值分三类：

①无连接关系：$F_{ij}=0$ 或 $P_{ij}=0$

②直接连接：$F_{ij}=1$ 或 $P_{ij}=1$

③通过放大系数为 k_{ij} 的放大环节连接：$F_{ij}=k_{ij}$ 或 $P_{ij}=k_{ij}$。当放大环节是反相器时，$k_{ij}=-1$。

应该指出，在环节编号时，要特别注意信号流动的方向。即环节编号应与信号流动方向一致，否则会引起信号滞后一拍或多拍的现象。

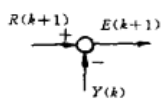


图 3－17　调节器的偏差计算

通常把控制器的比较部分作为下一拍的计算起点，如图 3－17 示，控制器偏差 $e(k+1)$ 由测量 $y(k)$ 与设定值 $R(k+1)$ 计算：

$$e(k+1) = R(k+1) - y(k) \tag{3-69}$$

根据系统的测量信号初值 $y(0)$，即可由（3－69）式计算下一拍的偏差值。并开始逐个环节计算直到求得 $y(1)$。如此递推，即可求得整个系统的输出响应。

为了减小误差，环节数不应分割过多。同时，可以加入校正补偿环节，例如加入环节 $\lambda e^{\nu T_s}$。

在数据输入时，由于连接矩阵 F 与 P 的元素大部分是零，因此，根据需输入非零元素的个数 F_D 与 P_D，可采用不同的方法。

当 $F_D < \frac{n \times nr}{3} - 1$ 或 $P_D < \frac{n \times n}{3} - 1$ 时，可输入 i, j 及 F_{ij} 或 P_{ij}。这里的 $F_{ij} \neq 0, P_{ij} \neq 0$。

当 $F_D \geqslant \frac{n \times nr}{3} - 1$ 或 $P_D > \frac{n \times n}{3} - 1$ 时，可按矩阵元素的输入方法输入 F_{ij} 及 P_{ij}，对于值为0的 F_{ij} 或 P_{ij}，则输入0，或直接输入（在用READ语句读数据时），或者输入0↵或直接↵（在用INPUT语句读数据时）。↵表示回车键。

3.5.2 双线性变换仿真程序

双线性变换是一种分式变换。采用计算机仿真时是用实系数多项式虚轴平移加上比例、颠倒系数等变换来实现的。

设连续系统用

$$G(s) = \frac{A_0(s)}{A_1(s)} = \frac{A(0,m)s^m + A(0,m-1)s^{m-1} + \cdots + A(0,1)s + A(0,0)}{A(1,n)s^n + A(1,n-1)s^{n-1} + \cdots + A(1,1)s + A(1,0)}$$

表示。$n \geqslant m$，则

$$s = \frac{2}{T_s}\frac{z-1}{z+1}$$

的双线性变换是通过下述的虚轴平移、比例、颠倒系数等变换逐步实现的。

令 $r_1 = \frac{z-1}{z+1}$，则 $s = \frac{2}{T_s} r_1$

$$G(s) = \frac{A_0(s)}{A_1(s)} = \frac{A_0\left(\frac{2}{T_s}r_1\right)}{A_1\left(\frac{2}{T_s}r_1\right)} = \frac{A0_1(r_1)}{A1_1(r_1)} \quad \text{（比例变换）}$$

令：$r_2 = \frac{2}{z+1}$ 则 $r_1 = -(r_2 - 1)$

$$G(z) = \frac{A0_1(r_1)}{A1_1(r_1)} = \frac{A0_1[-(r_2-1)]}{A1_1[-(r_2-1)]} = \frac{A0_2(r_2)}{A1_2(r_2)} \quad \text{（比例、平移变换）}$$

令：$r_3 = \frac{1}{z+1}$ 则 $r_2 = 2r_3$

$$G(z) = \frac{A0_2(r_2)}{A1_2(r_2)} = \frac{A0_2(2r_3)}{A1_2(2r_3)} = \frac{A0_3(r_3)}{A1_3(r_3)} \quad \text{（比例变换）}$$

令：$r_4 = z + 1$ 则 $r_3 = \frac{1}{r_4}$

$$G(z) = \frac{A0_3(r_3)}{A1_3(r_3)} = \frac{A0_3\left(\frac{1}{r_4}\right)}{A1_3\left(\frac{1}{r_4}\right)} = (z+1)^{n-m}\frac{A0_4(r_4)}{A1_4(r_4)} \quad \text{（颠倒系数）}$$

$$\frac{A0_4(r_4)}{A1_4(r_4)} = \frac{A0_4(z+1)}{A1_4(z+1)} = \frac{A2_1(z)}{A3_1(z)} \qquad \text{(平移变换)}$$

$$\therefore \quad G(z) = (z+1)^{n-m}\frac{A2_1(z)}{A3_1(z)} = \frac{A2(z)}{A3(z)}$$

$$= \frac{A(2,n)z^n + A(2,n-1)z^{n-1} + \cdots + A(2,1)z + A(2,0)}{A(3,n)z^n + A(3,n-1)z^{n-1} + \cdots + A(3,1)z + A(3,0)}$$

本仿真程序也可实现从 z 域到 s 域的变换。

〔例3－6〕 已知连续系统 $G(s) = \frac{s}{(s+1)^2}$，采用双线性变换，$T_s = 1$。$G(s)$ 写成 $G(s) = \frac{0s^2 + s + 0}{s^2 + 2s + 1}$

在打入 RUN ↵ 后，在 CRT 上显示：

BILINEAR TRAN SFORMATION

1 z Domain⇒ s Domain

2 s Domain⇒ z Domain

which is your choice (1 or 2)? 2

在? 后打入 2 并回车，则表示从 s 域到 z 域的双线性变换。并回答下列问题。

Input the order of system ? 2

Input the sampling time ? 1

A (0, 0) =? 0

A (0, 1) =? 1

A (0, 2) =? 0

A (1, 0) =? 1

A (1, 1) =? 2

A (1, 2) =? 1

在回答问题，即输入数据时，应打开打印机电源，并安装好打印纸。计算机即印出。

$$\frac{+1s}{1s^2 + 2s + 1} \Rightarrow \frac{2z^2 - 2}{9z^2 - 6z + 1}$$

如在? 后打入 1 并回车，则表示从 z 域到 s 域的双线性变换。同样要回答有关问题，这时的 A (0, 0)、A (1, 0) 等均是 z 多项式的系数。计算机经计算后也打印有关结果。

本程序规定系统的阶次 n 不超过 30，否则应修改 20 语句的定维值。程序如下：

```
10  REM Bilinear Transformation
20  DIM A (3, 30), B (3), T (30, 30)
30  CLS
40  LOCATE 2, 4
50  PRINT "B I L I N E A R    T R A N S F O R M A T I O N"
60  LOCATE 6, 1
70  PRINT "1 z Domain⇒s Domain"
80  LOCATE 8, 1
90  PRINT "2 s Domain⇒z Domain"
100  LOCATE 10, 1
```

```
110  INPUT "which is your choice (1 or 2)"; A$
120  A=VAL (A$) -1
130  IF A<0 OR A>1 THEN BEEP: GOTO 100
140  LOCATE 12, 1
150  INPUT "Input the order of system"; N
160  LOCATE 14, 1
170  INPUT "Input the sampling time"; TS
180  CLS
190  'REM Input the data
200  FOR I=0 TO 1
210  FOR J=0 TO N
220  LOCATE I*4+3, 2
230  PRINT "A ("; I; ","; J;
240  INPUT ") ="; A (I, J)
250  NEXT J
260  NEXT I
270  T (0, 0) =1
280  FOR I=1 TO N
290  FOR J=1 TO 1 STEP -1
300  T (0, J) =T (0, J) +T (0, J-1)
310  NEXT J
320  T (0, I) =1
330  T (I, 0) =1
340  NEXT I
350  FOR I=1 TO N
360  FOR J=1 TO N
370  T (I, J) =T (I-1, J) -T (I, J-1) -T (I-1, J-1)
380  NEXT J
390  NEXT I
400  K=2/TS
410  FOR I=0 TO N
420  A (2, I) =0
430  A (3, I) =0
440  NEXT I
450  FOR I=0 TO N
460  S=1
470  FOR J=0 TO N
480  FOR P=0 TO 1
490  A (P+2, I) =A (P+2, I) +S* (A (P, N-J) *A+A (P, J) * (A=
0)) *T (J, I)
```

```
500  NEXT P
510  S=S* (K*A- (A=0))
520  NEXT J
530  NEXT I
540  IF A<> 0 THEN 620
550  S=1
560  FOR J=N TO 0 STEP -1
570  FOR I=2 TO 3
580  A (I, J) =A (I, J) *S
590  NEXT I
600  S=S/K
610  NEXT J
620  FOR I=0 TO 3
630  B (I) =N*5
640  FOR J=0 TO N
650  B (I) =B (I) +LEN (STR$ (A (I, J)))
660  NEXT J
670  NEXT I
680  I=0
690  LPRINT CHR$ (27); "A"; CHR$ (10);
700  FOR L=II TO II+1
710  LPRINT "     ";
720  FOR J=0 TO N
730  GOSUB 1000
740  NEXT J
750  LPRINT
760  IF L=0 THEN LPRINT "……";: GOSUB 1500
770  IF L=2 THEN LPRINT "⇒…";: GOSUB 1500
780  NEXT L
790  IF II<>0 THEN END
800  II=2
810  A=1-A
820  LPRINT
830  LPRINT
840  GOTO 700
1000  REM Print Subroutine
1010  IF A (L, J) =0 THEN LPRINT "     ";
1020  IF A (L, J) <>0 THEN LPRINT A (L, J);
1030  IF A=0 AND A (L, J) <>0 AND J<>N THEN LPRINT "z";
1040  IF A=1 AND A (L, J) <>0 AND J<>N THEN LPRINT "s";
```

```
1050  LPRINT CHR$ (27); "S"; CHR$ (0);
1060  IF N-J=1 THEN 1080
1070  IF J<>N AND A (L, J) <>0 THEN LPRINT N-J;
1080  LPRINT CHR$ (27); "T";
1090  IF J<>N AND A (L, J+1) >0 THEN LPRINT "+";
1100  IF J<>N AND A (L, J+1) <0 THEN LPRINT " ";
1110  RETURN
1500  P=(ABS(B(L+1)-B(L))+B(L+1)+B(L))/2
1510  FOR I=0 TO P
1520  LPRINT "-";
1530  NEXT I
1540  LPRINT
1550  RETURN
```

双线性变换的离散化方法适用于用多项式形式表示的传递函数的转换。在采样周期不大时，有较高的精度，为了在某些特定频率有较高的精度，可以采用预畸的方法。

3.5.3 非线性系统数字仿真程序

本仿真程序共有12种类型的功能模块，其中非线性环节共5种，此外，还有一种纯滞后模块。功能模块一览表见表3-4。程序框图见图3-18。

程序中，变量说明如下：

N：组成系统的模块总数。本程序把非线性模块也作为一种模块处理，因此，共有非线性模块5种。纯滞后以类似方法处理。

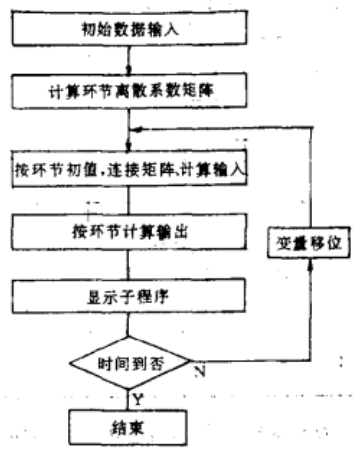
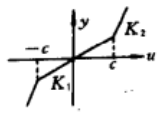
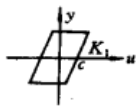
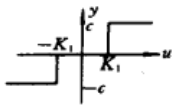
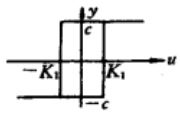
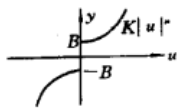


图3-18 非线性系统仿真程序框图

表3-4 功能模块一览表

类型号	传递函数或特性描述	离散系数阵			输入数据			
		$\Phi(T_s)$	$\Phi_m(T_s)$	$\Phi_m(T_s)$	$A1$	$A2$	$A3$	$A4$
1	$\frac{K}{s}$	1	KT_s	$\frac{KT_s^2}{2}$	K	0	0	0
2	$\frac{K}{Ts+1}$	$e^{-\frac{T_s}{T}}$	$K(1-e^{-\frac{T_s}{T}})$	$KT_s+KT(e^{-\frac{T_s}{T}}-1)$	K	T	0	0
3	$K\left(1+\frac{1}{T_s}\right)$	1	$\frac{K}{T_i}T_s$	$\frac{KT_s^2}{2T_i}$	K	T_i	0	0
4	$\frac{A_1}{s+\frac{1}{T_1}}+\frac{A_2}{s+\frac{1}{T_2}}$	$e^{-\frac{T_s}{T_1}}$ 0 0 $e^{-\frac{T_s}{T_2}}$	$T_1(1-e^{-\frac{T_s}{T_1}})$ $T_2(1-e^{-\frac{T_s}{T_2}})$	$T_1T_s+T_1^2(e^{-\frac{T_s}{T_1}}-1)$ $T_2T_s+T_2^2(e^{-\frac{T_s}{T_2}}-1)$	A_1	A_2	T_1	T_2
5	$\frac{K(s+b)}{s+a}$	e^{-aT_s}	$\frac{K}{a}(1-e^{-aT_s})$	$\frac{1}{a}\left[KT_s+\frac{K}{a}(e^{-aT_s}-1)\right]$	K	a	b	0
6	$\frac{cs+d}{s^2+as+b}$	$a_7+a_2a_6$ a_6/a_0 $-a_3a_6$ $a_7-a_2a_6$	$a_4(1-a_2a_6-a_7)$ $a_4(a_0^2+a_1^2/a_0)a_6$	a_8 a_9	a	b	c	d

续表

类型号	传递函数或特性描述	离散系数阵			输入数据			
		$\Phi(T_s)$	$\Phi_m(T_s)$	$\Phi_m(T_s)$	A1	A2	A3	A4
7					c	K_1	K_2	0
8					c	K_1	0	0
9					c	K_1	0	0
10					c	K_1	0	0
11					B	K	n	0
12	e^{-T_s}				T	0	0	0

注：$a_0=\sqrt{b-a^2/4}$；$a_1=a/2$；$a_2=a/2a_0$；$a_3=b/a_0$；$a_4=1/b$；$a_5=(Tb-a)/b^2$；$a_6=e^{-a_1T_s}\sin(a_0T_s)$；$a_7=e^{-a_1T_s}\cos(a_0T_s)$；$a_8=a_5+a_4^2[(a_1^2-a_0^2)a_6+2a_0a_1a_7]/a_0$；$a_9=a_4[1-(a_1^3/a_0+a_0a_1)a_4a_6-a_7]$

NR：输入变量个数。

TS：采样周期。

NB：打印间隔。它表示经 TS×NB 时间打印一次。因此，NB 应取整数 1，2，……。

TE：结束计算的时间。当用作图方式显示计算结果时，TE＝400×TS。其中，400 表示打印 400 点。

对于组成系统的各模块，应按下列顺序送入参数：

H（I），A1（I），A2（I），A3（I），A4（I），YO（I），UO（I），D（I）

其中，H（I）表示模块的类型号，见表 3－4。A1（I），A2（I），A3（I），A4（I）表示该模块的输入数据 A1、A2、A3、A4，见表 3－4。YO（I），UO（I）分别表示该模块的稳态输出和输入变量初值。D（I）是打印标志，当 D（I）项送入 1，表示打印输出该模块的输出值。送入其他数据则不打印出。

在使用本程序前，应先建立数据库。其次序是：P（I，J），F（I，J），Y1（I），UE（I），依次送入模块参数。P（I，J）是各模块间连接矩阵元素。F（I，J）是模块与外输入信号间连接矩阵元素。Y1（I）是各模块初始输出值。UE（I）是外输入信号的初值。值得注意的是模块参数的输入，必须依据模块的自然编号送入，而不是依据模块的类型号的次序。详见实例的应用。

〔例 3－7〕　如图 3－19 所示的非线性系统。求设定值变化到 10 时的系统输出响应。

该系统由5个模块组成。模块间连接关系是 $P=\begin{bmatrix}0&0&0&0&-1\\1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\end{bmatrix}$。模块有二个外输入信号 R 与 F。它们与模块的连接矩阵是 $F=\begin{bmatrix}1&-1\\0&0\\0&0\\0&0\\0&0\end{bmatrix}$。设定值 R 的初值是10，因此编入下列5000—5010语句。5个模块的参数如下：

模块编号	模块类型号	参数			
		A_1	A_2	A_3	A_4
Ⅰ	7	5	1	0	0
Ⅱ	5	1	0.2	0.5	0
Ⅲ	1	1	0	0	0
Ⅳ	2	1	0.5	0	0
Ⅴ	2	1	0.1	0	0

系统初值均为0。要求第Ⅴ号模块的输出显示。因此，相应的D（I）置1。综合上述，可以编制下列语句：

```
5000  DATA 0, 0, 0, 0, -1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0,
            0, 0, 0, 1, 0, 1, -1, 0, 0, 0, 0, 0, 0, 0, 0
5010  DATA 0, 0, 0, 0, 0, 10, 0
5020  DATA 7, 5, 1, 0, 0, 0, 0, 0
5030  DATA 5, 1, . 2, . 5, 0, 0, 0, 0
5040  DATA 1, 1, 0, 0, 0, 0, 0, 0
5050  DATA 2, 1, 0.5, 0, 0, 0, 0, 0
5060  DATA 2, 1, 0.1, 0, 0, 0, 0, 1
```

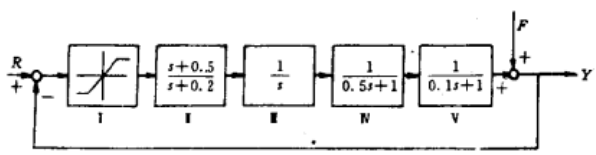


图3-19　非线性控制系统框图

系统在执行RUN命令后，进行人机对话。

计算机显示：The Number of Blocks

键入5，并回车。

显示：The Number of Input Variables

键入2，并回车。

显示：The Sampling Time

键入 0.05，并回车。

显示：The Interval of Printing

键入 10，并回车。

显示：The End Time

键入 10，并回车。

显示：1：CRT Display　2：Draw Curve　3：Print Data Select 1，2 or 3 for Display

根据显示方式，可以键入 1，2 或 3，并回车，表示显示数据在 CRT、绘制曲线或打印数据。

若键入 2，并回车。则还需回答范围最大与最小的值，这是指输出坐标的最大与最小值。本例中，可以键入 20 及 0，并回车。则先在 CRT 上绘出输出响应曲线，同时在画面下方显示每计算 10 个采样值后的数据。最后在打印机准备好的条件下打印出如下响应曲线。

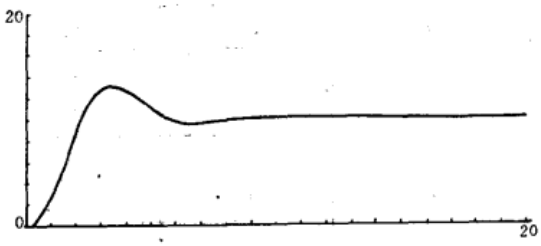


图 3－20　非线性控制系统输出响应曲线

〔例 3－8〕　如图 3－21 所示纯滞后系统，求设定值变化到 1 时的输出响应。

如图 3－21 所示，该系统有两个线性环节及一个纯滞后环节组成。输入的语句是：

```
5000  DATA 0, 0, -1, 1, 0, 0,
0, 1, 0, 1, -1, 0, 0, 0, 0
5010  DATA 0, 0, 0, 1, 0
5020  DATA 3, 8, 3.3, 0, 0, 0,
0, 1
5030  DATA 2, 0.5, .5, 0, 0, 0, 0, 1
5040  DATA 12, 1, 0, 0, 0, 0, 0, 1
```

人机对话时，分别键入 3；2；0.05；10；10，并在显示方式选择时，选显示数据打印。则有下列表格会被印出。

Simulation Report

Sampling Time Ts＝.05

Block No.1 G(s)＝K＊(1＋1/Tis)

Block Parameter K＝8 Ti＝3.3

Block No. 2 G(s)＝K/(Ts＋1)

Block Parameter K＝.5 T＝5

Block No. 3 Dead time T

Block Parameter T＝1

Time	Y (1)	Y (2)	Y (3)
0	8.182	0	0
.5	9.394	.437	0
1	10.606	.872	0
1.5	8.391	1.248	.437
2	5.348	1.457	.872
2.5	2.14	1.492	1.248
3	－.121	1.393	1.457

3.5	−1.128	1.225	1.492
4	−.965	1.054	1.393
4.5	−.027	.929	1.225
5	1.189	.868	1.054
5.5	2.258	.869	.929
6	2.925	.912	.868
6.5	3.125	.971	.869
7	2.943	1.024	.912
7.5	2.545	1.058	.971
8	2.11	1.067	1.024
8.5	1.772	1.058	1.058
9	1.597	1.036	1.067
9.5	1.586	1.013	1.058
10	1.693	.994	1.036

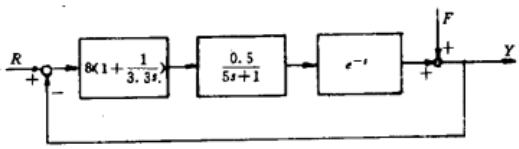


图 3－21　纯滞后控制系统框图

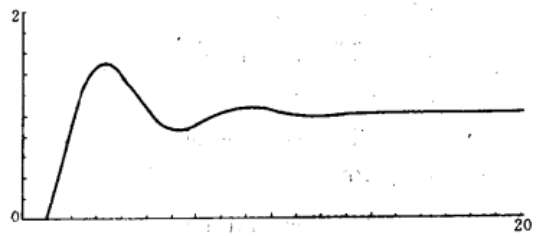


图 3－22　纯滞后控制系统的输出响应曲线

若在显示方式中选择2，即绘制曲线。则可以得到与图2－20相似的曲线。如下图所示。

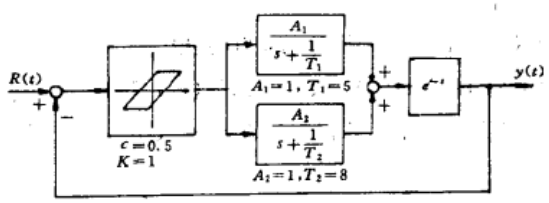


图 3－23　滞环非线性系统

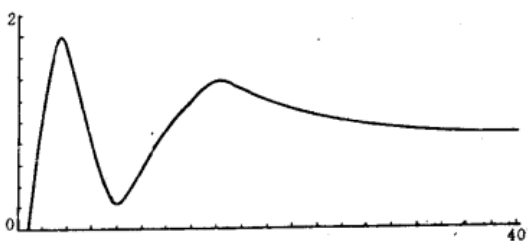


图 3－24　闭环输出响应

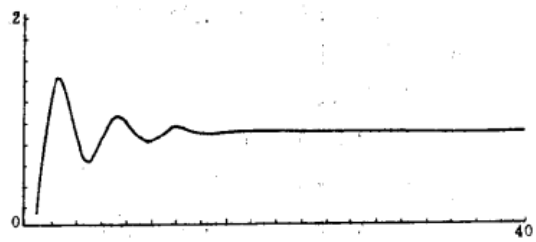


图 3－25　线性系统输出

在绘制闭环输出响应时，5020，5030 语句中最后的数字 1 应改为 0，表示该两个模块不

必输出响应曲线。

〔例 3－9〕　对图 3－23 所示非线性系统，求系统输出响应曲线。

根据表 3－4，该系统由 3 个模块组成。类型号 8、4 和 12。输入相应语句后，得输出响应曲线如图 3－24。

可以看到，由于系统不含积分环节，因此，存在余差。而滞环特性使控制品质变差。当滞环用线性环节代替，得到图 3－25 所示的系统响应。可见控制品质明显改善。

程序清单如下。程序中的离散化方法采用状态变量法。

```
4  REM Simulation Program Ⅳ, Copyright by ECUCT, 1985, 1986
6  CLS
8  LOCATE 2, 1
10  PRINT "Nonlinear System Simulation"
12  LOCATE 4, 1
14  INPUT"The Number of Blocks "; N
16  LOCATE 6, 1
18  INPUT"The Number of Input Variables"; NR
20  LOCATE 8, 1
22  INPUT"The Sampling Time "; TS
24  LOCATE 10, 1
26  INPUT"The Interval of Printing "; NB
28  LOCATE 12, 1
30  INPUT"The End Time "; TE
32  LOCATE 14, 1
34  PRINT " 1: CRT Display 2: Draw Curve 3: Print Data": PRINT
36  INPUT"Select 1, 2 or 3 for Display "; AY
38  PR = VAL(AY) - 1
40  IF PR < 0 OR PR > 2 THEN BEEP: GOTO 32
42  DIM A(N, 4), B(N, 4), Y1(N), Y0(N), U1(N), U0(N), F(N, NR), P(N, N)
44  DIM A1(N), A2(N), A3(N), A4(N), H(N), UE(NR), UF(N), D(N), X1(N), X2
    (N), G(N, 1000)
50  FOR I = 1 TO N
52  FOR J = 1 TO N
54  READ P(I, J): REM BLOCK LINK MATRIX ELEMENTS
56  NEXT J
58  NEXT I
60  FOR I = 1 TO N
62  FOR J = 1 TO NR
64  READ F(I, J): REM INPUT LINK MATRIX ELEMENTS
66  NEXT J
68  NEXT I
70  FOR I = 1 TO N
```

```
72  READ Y1(I):REM INIT VALUE
74  NEXT I
76  FOR I=1 TO NR
78  READ UE(I):REM EXTERNAL INPUT
80  NEXT I
82  DN=0
84  FOR I=1 TO N
86  READ H(I), A1(I), A2(I), A3(I), A4(I), Y0(I), U0(I), D(I)
87  REM BLOCK No., Para(4), Init Val(2), Print Flag
88  IF D(I)=1 THEN DN=DN+1
90  NEXT I
92  FOR I=1 TO N
94  ON H(I) GOSUB 1010, 1020, 1030, 1040, 1050, 1060, 1080, 1080, 1080, 1080, 1080,
    1080
96  NEXT I
98  IF PR=0 THEN GOSUB 3200
100  T=0:BN=NB
102  IF PR=1 THEN GOSUB 3000
104  IF PR=2 THEN GOSUB 3500
106  FOR I=1 TO N
108  U=0
110  FOR J=1 TO N
112  U=U+P(I,J)*Y1(J)
114  NEXT J
116  FOR J=1 TO NR
118  U=U+F(I,J)*UE(J)
120  NEXT J
122  U1(I)=U
123  NEXT I
124  FOR I=1 TO N
126  UF(I)=(U1(I)-U0(I))/TS
128  NEXT I
130  FOR I=1 TO N
132  ON H(I) GOSUB 2000, 2000, 2000, 2000, 2000, 2000, 2070, 2080, 2090, 2100, 2110,
    2120
133  UO(I)=U1(I):Y0(I)=Y1(I)
134  NEXT I
136  ON PR+1 GOSUB 3218, 3100, 3300
138  T=T+TS
140  Z=Z+1
```

```
142  IF Z>400 THEN GOSUB 4000
144  IF T>TS+TE THEN END
146  GOTO 106
1010  X1(I)=Y1(I)
1011  A(I,1)=1
1012  A(I,2)=TS*A1(I)
1013  A(I,3)=A(I,2)*TS/2
1014  RETURN
1020  X1(I)=Y1(I)
1021  A(I,1)=EXP(-TS/A2(I))
1022  A(I,2)=A1(I)*(1-A(I,1))
1023  A(I,3)=A1(I)*TS-A(I,2)*A2(I)
1024  RETURN
1030  X1(I)=Y1(I)
1031  A(I,1)=1
1032  A(I,2)=TS*A1(I)/A2(I)
1033  A(I,3)=A(I,2)*TS/2
1034  RETURN
1040  X1(I)=Y1(I)/(A1(I)*A3(I)+A2(I)*A4(I))*A3(I)
1041  X2(I)=Y1(I)/(A1(I)*A3(I)+A2(I)*A4(I))*A4(I)
1042  A(I,1)=EXP(-TS/A3(I))
1043  A(I,2)=(1-A(I,1))*A3(I)
1044  A(I,3)=(TS-A(I,2))*A3(I)
1045  A(I,4)=0:B(I,1)=0
1046  B(I,4)=EXP(-TS/A4(I))
1047  B(I,2)=A4(I)*(1-B(I,4))
1048  B(I,3)=A4(I)*(TS-B(I,2))
1049  RETURN
1050  X1(I)=Y1(I)/A3(I)
1051  A(I,1)=EXP(-TS*A2(I))
1052  A(I,2)=A1(I)*(1-A(I,1))/A2(I)
1053  A(I,3)=(TS*A1(I)-A(I,2))/A2(I)
1054  RETURN
1060  X1(I)=Y1(I)/A4(I)
1061  X2(I)=0
1062  A0=SQR(A2(I)-A1(I)*A1(I)/4)
1063  A1=A1(I)/2
1064  A2=A1(I)/A0/2
1065  A3=A2(I)/A0
1066  A4=1/A2(I)
```

```
1067  A5=(TS*A2(I)-A1(I))/A2(I)/A2(I)
1068  A6=EXP(-A1*TS)*SIN(A0*TS)
1069  A7=EXP(-A1*TS)*COS(A0*TS)
1070  A(I,1)=A7+A2*A6
1071  A(I,4)=A6/A0
1072  A(I,2)=A4*(1-A2*A6-A7)
1073  A(I,3)=A5+A4*A4*((A1*A1-A0*A0)*A6+2*A0*A1*A7)/A0
1074  B(I,1)=-A3*A6
1075  B(I,4)=A7-A2*A6
1076  B(I,2)=A4*(A0+A1*A1/A0)*A6
1077  B(I,3)=A4*(1-(A1*A1*A1/A0+A0*A1)*A4*A6-A7)
1078  RETURN
1080  RETURN
2000  X1(I)=A(I,1)*X1(I)+A(I,4)*X2(I)+A(I,2)*U1(I)+A(I,3)*UF(I)
2002  IF H(I)<=2 THEN Y1(I)=X1(I)
2004  IF H(I)=3 THEN Y1(I)=X1(I)+A1(I)*U1(I)
2006  X2(I)=B(I,4)*X2(I)+B(I,1)*X1(I)+B(I,2)*U1(I)+B(I,3)*UF(I)
2008  IF H(I)=4 THEN Y1(I)=A1(I)*X1(I)+A2(I)*X2(I)
2010  IF H(I)=5 THEN Y1(I)=(A3(I)-A2(I))*X1(I)+A1(I)*U1(I)
2012  IF H(I)=6 THEN Y1(I)=A4(I)*X1(I)+A3(I)*X2(I)
2014  RETURN
2070  Y1(I)=A2(I)*U1(I)
2071  IF ABS(U1(I))>A1(I) THEN Y1(I)=SGN(U1(I))*(A2(I)-A3(I))*A1(I)
      +A3(I)*U1(I)
2072  RETURN
2080  AA=SGN(U1(I)-U0(I))
2081  Y1(I)=(U1(I)-AA*A1(I))*A2(I)
2082  IF Y1(I)*AA<Y0(I)*AA THEN Y1(I)=Y0(I)
2083  Y0(I)=Y1(I)
2084  U0(I)=U1(I)
2085  RETURN
2090  Y1(I)=SGN(U1(I))*A1(I)
2091  IF ABS(U1(I))<=A2(I) THEN Y1(I)=0
2092  RETURN
2100  AA=SGN(U1(I)-U0(I))
2101  Y1(I)=AA*A1(I)
2102  IF ABS(U1(I))<A2(I) THEN Y1(I)=Y0(I)
2103  Y0(I)=Y1(I)
2104  U0(I)=U1(I)
2105  RETURN
```

```
2110  Y1(I) = (A2(I) * ABS(U1(I))^A3(I) + A1(I)) * SGN(U1(I))
2111  RETURN
2120  A1 = A1(I)/TS
2121  G(I, INT(A1)) = U1(I)
2122  FOR J = 1 TO INT(A1)
2123  G(I, J - 1) = G(I, J)
2124  NEXT J
2125  A2 = A1 - INT(A1)
2126  Y1(I) = G(I, 0) * (1 - A2/TS) + G(I, 1) * A2/TS
2127  RETURN
3000  CLS
3002  LOCATE 2, 1
3004  INPUT "The Range Max "; RMAX
3006  LOCATE 4, 1
3008  INPUT "The Range Min "; RMIN
3010  CS = 100/(RMAX - RMIN)
3012  NB = BN
3014  CLS
3016  SCREEN 2, 1
3018  LINE (80, 10) - (80, 110)
3020  LINE - (480, 110)
3022  FOR I = - 1 TO 9
3024  PSET (81, I * 10 + 20)
3026  NEXT I
3028  FOR I = 1 TO 20
3030  PSET (I * 20 + 80, 109)
3032  FOR J = 0 TO 9
3034  PSET (I * 20 + 80, J * 10 + 10)
3036  NEXT J
3038  NEXT I
3040  LOCATE 16, 3
3042  PRINT"Time"
3044  B = 0
3046  FOR I = 1 TO N
3048  IF D(I) = 0 THEN 3054
3050  LOCATE 16, B * 60/DN + 9
3052  PRINT"Y("; I; ")"
3053  B = B + 1
3054  NEXT I
3056  LOCATE 2, 1
```

```
3058  PRINT SPC(8 - LEN(STRY(RMAX)));RMAX
3060  LOCATE 14, 1
3062  PRINT SPC(8 - LEN(STRY(RMIN)));RMIN
3064  LOCATE 15, 52
3066  RH = INT(TS * 400 * 100)/100
3068  PRINT SPC(10 - LEN(STRY(RH)));RH
3070  TE = 400 * TS
3072  PN = 15
3074  RETURN
3100  B = 0
3102  FOR I = 1 TO N
3104  IF D(I) = 0 THEN 3108
3106  PSET(Z + 80, 110 - CS * (Y1(I) - RMIN))
3108  NEXT
3110  REM PN = 14
3112  GOSUB 3218
3114  RETURN
3200  CLS: LOCATE 1, 1
3202  PRINT"Time"
3204  B = 0
3206  FOR I = 1 TO N
3208  IF D(I) = 0 THEN 3214
3210  LOCATE 1, 60/DN * B + 10
3212  PRINT"Y(";I;")"
3213  B = B + 1
3214  NEXT I
3216  PN = 0 :RETURN
3218  IF BN< >NB THEN 3256
3220  LOCATE 3 + PN, 1
3222  DT = INT(T * 1000 + .5)/1000
3224  PRINT DT
3226  LOCATE 4 + PN, 1
3228  IF PN = 19 AND PR = 0 THEN LOCATE 3, 1
3229  IF PN = 19 AND PR = 1 THEN LOCATE 18, 1
3230  PRINT"          "
3232  B = 0
3234  FOR I = 1 TO N
3236  IF D(I) = 0 THEN 3250
3238  LOCATE 3 + PN, 8 + B * 60/DN
3240  PRINT INT(Y1(I) * 1000 + .5)/1000
```

```
3242  LOCATE 4 + PN, 8 + B * 60/DN
3245  IF PN = 19 AND PR = 1 THEN LOCATE 18, 8 + B * 60/DN
3244  IF PN = 19 AND PR = 0 THEN LOCATE 3, 8 + B * 60/DN
3246  PRINT"      "
3248  B = B + 1
3250  NEXT I
3252  PN = PN + 1
3254  BN = 0
3256  BN = BN + 1
3258  IF PN > 19 AND PR = 0 THEN PN = 0
3260  IF PN > 19 AND PR = 1 THEN PN = 15
3262  RETURN
3300  IF BN < > NB THEN 3322
3302  B = 0
3304  DT = INT(T * 1000 + .5)/1000
3306  LPRINT DT;
3308  FOR K = 1 TO N
3310  IF D(K) = 0 THEN 3316
3312  LPRINT TAB (60 * B/DN + 8); Y1(K);
3314  B = B + 1
3316  NEXT K
3318  LPRINT
3320  BN = 0
3322  BN = BN + 1
3324  RETURN
3500  LPRINT
3502  LPRINT"Simulation Report"
3504  LPRINT
3506  LPRINT"Sampling Time Ts = "; TS
3508  LPRINT
3510  FOR I = 1 TO N
3512  ON H(I) GOSUB 3610, 3620, 3630, 3640, 3650, 3660, 3670, 3680, 3690, 3700,
      3710, 3720
3514  NEXT
3516  LPRINT "Time";
3518  B = 0
3520  FOR K = 1 TO N
3522  IF D(K) = 0 THEN 3528
3524  LPRINT TAB(60 * B/DN + 10); "Y("; K; ")";
3526  B = B + 1
```

```
3528  NEXT K
3530  LPRINT
3532  RETURN
3610  LPRINT "Block No.";I;" G(s)=K/s"
3611  LPRINT "Block Parameter K=";A1(I)
3612  LPRINT
3613  RETURN
3620  LPRINT "Block No.";I;"G(s)=K/(Ts+1)"
3621  LPRINT "Block Parameter K=";A1(I);" T=";A2(I)
3622  LPRINT
3623  RETURN
3630  LPRINT "Block No.";I;"G(s)=K*(1+1/Tis)"
3631  LPRINT "Block Parameter K=";A1(I);" Ti=";A2(I)
3632  LPRINT
3633  RETURN
3640  LPRINT "Block No.";I;" G(s)=A1*T1/(T1s+1)+A2*T2/(T2s+1)"
3641  LPRINT "Block Parameter A1= ";A1(I);" A2=";A2(I)
3642  LPRINT "Block Parameter T1= ";A3(I);" T2=";A4(I)
3643  LPRINT
3644  RETURN
3650  LPRINT "Block No.";I;" G(s)=K*(s+b)/(s+a) "
3651  LPRINT "Block Parameter K=";A1(I)
3652  LPRINT "Block Parameter a=";A2(I);"b=";A3(I)
3653  LPRINT
3654  RETURN
3660  LPRINT "Block No.";I;" G(s)=cs+d/(s*s+as+b)"
3661  LPRINT "Block Parameter a=";A1(I);" b=";A2(I)
3662  LPRINT "Block Parameter c=";A3(I);" d=";A4(I)
3663  LPRINT
3664  RETURN
3670  LPRINT "Block No. ";I;" Nonlinear with Multilinear Form"
3671  LPRINT "Block Parameter c=";A1(I);" k1=";A2(I)
3672  LPRINT "Block Parameter k2=";A3(I)
3673  LPRINT
3674  RETURN
3680  LPRINT "Block No. ";I;"Hysteresis Loop Nonlinear"
3681  LPRINT "Block Parameter c=";A1(I);" k1=";A2(I)
3682  LPRINT
3683  RETURN
3690  LPRINT "Block No. ";I;"Relay Nonlinear with Dead Zone"
```

```
3691  LPRINT "Block Parameter c=";A1(I);" k1=";A2(I)
3692  LPRINT
3693  RETURN
3700  LPRINT "Block No. ";I;"Hysteresis Loop Nonlinear with Relay"
3701  LPRINT "Block Parameter c=";A1(I);" k1=";A2(I)
3702  LPRINT
3703  RETURN
3710  LPRINT "Block No. ";I;"Nonlinear with Variable Gain"
3711  LPRINT "Block Parameter B=";A1(I);" k=";A2(I)
3712  LPRINT "Block Parameter n=";A3(I)
3713  LPRINT
3714  RETURN
3720  LPRINT"Block No.";I;" Dead time T"
3721  LPRINT "Block Parameter T=";A1(I)
3722  LPRINT
3723  RETURN
4000  FOR I=1 TO 7
4010  LOCATE I+15,1
4020  FOR J=1 TO 80
4030  PRINT"  ";
4040  NEXT J
4050  NEXT I
4060  KEY OFF
4070  LPRINT CHRY(27)"W"
4080  RETURN
```

4 采样控制系统的数字仿真

4.1 采样控制系统

4.1.1 概述

4.1.1.1 采样控制系统

采样控制系统是一类控制系统，在这类系统中，有一个或多个变量仅在采样的瞬时才发生变化，即只能在采样时刻才能对这些变量进行测量或由数字计算机的存贮器中进行读出。用数字计算机或数字控制器来进行控制的系统是典型的采样控制系统。图 4－1 是采样控制系统框图。

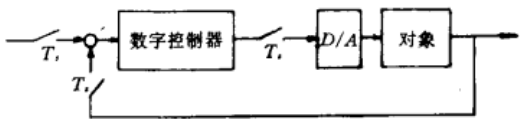


图 4－1 采样控制系统框图

根据控制器的类型，采样控制系统分为两类：模拟控制器采样控制系统和数字控制器采样控制系统。如图 4－2 所示。数字计算机控制系统是采用数字控制算法的采样控制系统。随着数字计算机在控制系统中的广泛应用，对它们进行分析、研究和数字仿真是十分必要的。

4.1.1.2 采样控制系统的特点

采样控制系统数字仿真的特点是：

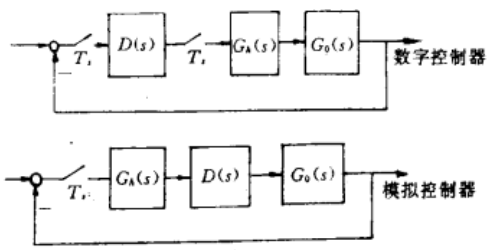


图 4－2 采样控制系统的类型

①其数字仿真的方法和第三章所述的连续系统离散化数字仿真方法相同。所不同的是：连续系统离散化模型是人为地加入虚拟的采样开关和保持器，而采样控制系统则本身存在采样开关和保持器。因此，连续系统离散化模型是一个近似模型，而采样控制系统中对象本身往往是连续的，只是由于加了采样开关，才呈现时间离散特性。因此，可直接建立差分方程的数学模型。

②采样控制系统通常由两部分组成。一部分是连续的被控对象，另一部分是离散的数字控制器。这两部分的采样周期可以采用不同的数值。

③由于采样的原因，连续部分的输出只能在采样瞬间测量，在两个采样周期之间的输出不能直接获得。采样周期较小时，可用内插法线性插值求得，或用改进 z 变换法求出。

④采样控制系统采用的 D/A 转换器相当于零阶保持器，它能保证采样间隔内的信号为恒定值。因此，采样控制系统中，对于连接在数字控制器输出端的连续部分来说，相当于在采样瞬间加入阶跃信号。

4.1.1.3 采样控制系统仿真时应注意的问题

由于采样控制系统的特点，在数字仿真时，应注意下列问题。

①采样控制系统由两部分组成。这两部分的离散化方法可以不同，采样周期或积分步长也可不同，因此，在仿真时为了两部分的衔接，采样控制系统实际存在的采样开关的采样周期 T_s 应是系统中连续部分采样周期或积分步长 H 的整数倍。$T_s = NH$，$N = 1, 2\cdots$

②若连续部分可直接得到差分方程，而且不要求计算中间的状态变量，为简化数字仿真，可用相同的采样周期 T_s，分别列写离散与连续部分的差分方程，然后进行数字仿真。若连续部分不易直接获得差分方程，或者存在非线性环节，或者需计算中间状态变量时，常采用对连续部分用 H 积分步长作 N 次计算，然后对离散部分作一次计算的仿真方法。

③为了得到两个采样时刻之间的响应，需要采用改进 z 变换法。仿真的算法上，由于改进 z 变换法引入了虚拟的滞后或超前环节，因此在数值的处理方法上有所不同。

4.1.2 改进 z 变换

4.1.2.1 改进 z 变换的定义和求法

改进 z 变换是指系统中含有纯滞后环节时所求得的 z 变换。它是对 z 变换用虚拟时间滞后或超前的方法进行修正。

假设一个具有纯滞后为 τ 的系统，其传递函数如下：

$$G_p(s) = G(s) \cdot e^{-\tau s} \tag{4-1}$$

式中，$G(s)$ 是不含纯滞后的对象传递函数。离散化时采用的采样周期是 T_s，则有：

$$\tau = NT_s + \theta$$

式中，$N = \mathrm{INT}\left(\frac{\tau}{T_s}\right)$，表示在 τ 中所包含的 T_s 的最大整数个数。显然，$0 \leqslant \theta < T_s$。

因此，式（4-1）成为：

$$G_p(s) = G(s) \cdot e^{-(NT_s+\theta)s} \tag{4-2}$$

当对式（4-2）进行 z 变换时，有：

$$G_p(z) = Z[G(s)e^{-(NT_s+\theta)s}] = z^{-N}Z[G(s)e^{-\theta s}] \tag{4-3}$$

则改进 z 变换定义为 $G(z, m)$：

$$G(z, m) = Z_m[G(s)] = Z[G(s)e^{-\theta s}] \tag{4-4}$$

改进 z 变换的求法和 z 变换的求法相同。常用的有无穷级数求和及拉氏变换直接求出（求留数）法。典型环节的改进 z 变换，也可通过查表获得。

令 $\theta = (1-m)T_s$　　则 $0 < m < 1$

$$G(z, m) = Z[G(s) \cdot e^{-(1-m)T_s s}] = z^{-1}Z[e^{mT_s s}G(s)]$$

这表明，通过一定的 m 值的设置，能够获得在两个采样值间的函数值。

若 $G(s)$ 的权函数是 $g(t)$，根据控制原理的有关知识可得纯滞后改进 z 变换的表达式：

$$G(z, m) = z^{-1}\sum_{n=0}^{\infty} g(nT_s + mT_s)z^{-n} \tag{4-5}$$

当系统中串接一个超前环节，设超前时间 $\theta = \Delta T_s, (0 < \Delta < 1)$，同样，可以求得超前改进 z 变换的表达式：

$$G(z, \Delta) = \sum_{n=0}^{\infty} g(nT_s + \Delta T_s)Z^{-n} = Z[e^{\Delta T_s s}G(s)] \tag{4-6}$$

通过留数的计算，由拉氏变换求得改进 z 变换的计算公式如下：

$$G(z,m)=z^{-1}\sum_i R_{es}\left[\frac{G(p)e^{mT_sp}}{1-e^{T_sp}z^{-1}}\right]_{p=s_i} \tag{4-7}$$

〔例 4－1〕　求 $G(s)=\dfrac{1}{(s+a)^2}$ 的改进 z 变换。

根据（4－6），可得：

$$\begin{aligned}G(z,m)&=z^{-1}\frac{d}{dp}\left[(p+a)^2\cdot\frac{1}{(p+a)^2}\cdot\frac{e^{mT_sp}}{1-e^{T_sp}z^{-1}}\right]\Bigg|_{p=-a}\\&=z^{-1}\frac{mT_s(1-e^{T_sp}z^{-1})e^{mT_sp}+e^{mT_sp}T_se^{T_sp}z^{-1}}{(1-e^{T_sp}z^{-1})^2}\Bigg|_{p=-a}\\&=\frac{T_se^{-amT_s}[e^{-aT_s}+m(z-e^{-aT_s})]}{[z-e^{-aT}]^2}\end{aligned}$$

4.1.2.2　应用改进 z 变换计算闭环系统的响应

下面的实例说明了应用改进 z 变换计算闭环系统响应的方法。

〔例 4－2〕　某具有纯滞后 $\tau=1.25$ 的采样控制系统如图 4－3 所示。采样周期 $T_s=1$。计算在 a 点的输出响应。

$\tau=1.25$ 可分解为整数和小数两部分：

$\tau=T_s+0.25T_s$；　$\delta=0.25$

引入虚拟的超前环节，令：

$m_\delta=1-\delta=0.75$；　$\lambda=2-m$；　$\Delta=m$

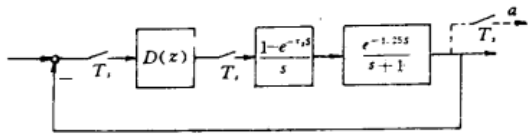


图 4－3　纯滞后采样控制系统

从图 4－4 可以看到，当 Δ 从 0 变化到 1 时，T_s 时刻 c 点的响应相当于 a 点从 $(1+\delta)T_s$ 到 $(2+\delta)T_s$ 这一段时刻的响应。可以看到，当 $\Delta=m_\delta$ 时，在 T_s 时刻 c 点的响应就是 a 点在 $2T_s$ 时刻的响应。据此，即求得各采样时刻间的输出。

计算对象的改进 z 变换：

$$G(z,m)=z^{-1}Z_m\left[\frac{1-e^{-T_ss}}{s(s+1)}\right]=\frac{z-1}{z^2}\left(\frac{1}{z-1}-\frac{e^{-m}}{z-0.368}\right)$$

$$G(z,\Delta)=zZ_m\left[\frac{1-e^{-T_ss}}{s(s+1)}\right]\Bigg|_{m=\Delta}=\frac{z-1}{z^2}\left(\frac{1}{z-1}-\frac{e^{-\Delta}}{z-0.368}\right)$$

$m_\delta=0.75$，∴有：

$$G(z,m_\delta)=G(z,m)\mid_{m=m_\delta}=\frac{0.527z+0.105}{z(z-0.368)}$$

闭环系统的脉冲传递函数：

$$\begin{aligned}\frac{Y(z,\Delta)}{R(z)}&=\frac{G(z,\Delta)}{1+G(z,m_\delta)}\\&=\frac{(1-e^{-\Delta})z^2+(e^{-\Delta}-0.368)z}{z^3-0.368z^2+0.527z+0.105}\end{aligned}$$

差分方程是：

$$\begin{aligned}y(h,\Delta)=&0.368y(k-1,\Delta)-0.527y(k-2,\Delta)\\&+0.105y(k-3,\Delta)+(1-e^{-\Delta})R(k-1)+(e^{-\Delta}-0.368)R(k-2)\end{aligned}$$

当 $k=1$，2，3…，$\Delta=m_\delta=0.75$ 时，可以得到 a 点在 2，3，4，…倍采样时刻时的响应。当 Δ 从 0 到 1 变化，则可求得 a 点在 $(k+\delta)T_s$ 到 $(k+1+\delta)T_s$ 这段时间的输出响应。

当需要求出数字控制器 $D(z)$ 的输入和输出信号时，可以将控制器 $D(z)$ 的差分方程先计算，求出它的输出并作为连续部分的输入。采样控制系统常用零阶保持器，因此在计算连续部分的过程中，它的输入保持不变，直到本采样周期结束，并得到下一采样周期时控制器的输入，如此循环迭代，直到按要求求出所需的全部响应值。

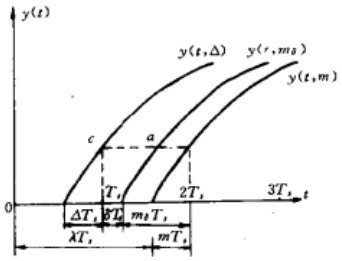


图 4－4　采样时刻之间的输出响应

可以看到，由于在差分方程中含有变量 Δ，它不仅需要知道前一步的数据，还要知道前几步的数据，所以在数据处理时，要对每一步的以 Δ 为步距的数据设置存放单元，而且要对前几步的相应数组设置存放单元。以便于数据的存放、平移等处理过程的进行。

4.1.3　数字控制器的描述

4.1.3.1　控制规律的描述

与模拟控制器的控制规律相类似，数字控制器也有比例（P）、比例积分（PI）、比例积分微分（PID）、比例微分（PD）、测量微分等控制规律。

实施时，常有位置式、增量式及速度式三种算法。如图 4－5 所示，数字控制器的输入 $e(t)$ 经采样开关后送入，经控制算法的计算，其输出经采样开关后送到保持器，经保持后送到广义对象去。由于数字计算机采用了较高的运算速度，可以认为两个采样开关是同步运行的。

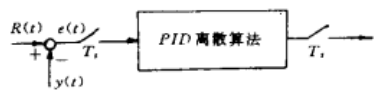


图 4－5　PID 控制规律的仿真

位置式算法的计算公式如下：

$$u(k) = K_c e(k) + \frac{K_c}{T_i}\sum_{i=0}^{k} e(i)T_s + K_c T_d \frac{e(k) - e(k-1)}{T_s}$$

$$= K_c e(k) + K_I \sum_{i=0}^{k} e(i) + K_D[e(k) - e(k-1)] \tag{4-8}$$

式中：　$K_I = \dfrac{K_c T_s}{T_i}$；$K_D = \dfrac{K_c T_d}{T_s}$

增量式算法的计算公式如下：

$$\begin{aligned}\Delta u(k) &= u(k) - u(k-1)\\ &= K_c[e(k) - e(k-1)] + K_I e(k)\\ &\quad + K_D[e(k) - 2e(k-1) + e(k-2)]\end{aligned} \tag{4-9}$$

速度式算法的计算公式如下：

$$v(k) = \frac{\Delta u(k)}{T_s} = K_c \frac{\Delta e(k)}{T_s} + \frac{K_c}{T_i} e(k)$$

$$+ \frac{K_c T_d}{T_s^2}[e(k) - 2e(k-1) + e(k-2)] \tag{4-10}$$

在计算机实施上述算法时，应注意几点改进方法：

①解决积分控制与字长的矛盾：在积分控制作用式中，$\Delta u_I(k) = K_I e(k)$，当机器字长一定时，若 $\Delta u_I(k)$ 小于某一阈值，则计算机就认为 $\Delta u_I(k)$ 是机器零，此时增量式的积分输出项为零，而实际偏差 $e(k)$ 存在，为解决这两者的矛盾，要加入 $\sum e\ (i)$ 的存贮单元，用于在 $\Delta u_I\ (k)$ 为机器零时，把 $e\ (i)$ 进行累加，直到 $\Delta u_I\ (k+j)$ 不为机器零时，把此

时的 $\Delta u_I(k+j)$ 作为积分项的输出，同时把累加单元 $\Sigma e(i)$ 清零，从而消除余差。

②积分分离：加入积分作用的目的是消除余差，因此，根据是否有利于消除余差，对不同的场合，采用加入或不加入积分作用的积分分离算法是有用的。积分分离的方法有两种。

i. 如图 4－6 所示，当比例作用输出 u_p 与积分作用输出同方向时，应引入积分作用，反之则应切除积分作用。

其子程序如下

```
1000  U(I) = KC * E(I)
1002  UI(I) = UI(I) + KI * E(I)
1004  IF U(I) * UI(I) > 0 THEN U(I) = U(I) + UI(I)
1006  RETURN
```

程序中，U（I）是 PI 调节器输出，I 是步数，UI（I）是积分项输出。

ii. 积分作用仅在偏差小于某限值时才加入。子程序只需把 1004 语句改为：

```
1004  IF E(I) < E  THEN U(I) = U(I) + UI(I)
```

程序中，E 即是切入积分作用的限值。

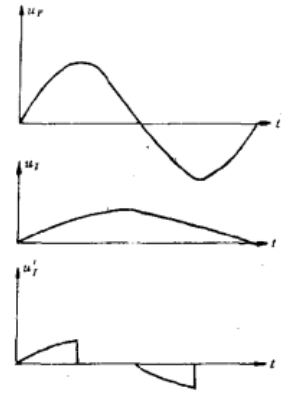


图 4－6　积分分离的算法之一

③测量微分：微分作用是用于克服滞后，当 $e(k)$ 发生突变时，会引起微分作用的输出项变得很大，为此，可采用测量微分，它是对测量值 $y(k)$ 进行微分运算，而使设定值变化时，控制器输出不会突变，而被控变量的变化是平缓的，因此在采样控制系统的数字控制算法中得到广泛应用。其算式是：

$$\Delta u_D(k) = -K_D[y(k) - 2y(k-1) + y(k-2)] \qquad (4-11)$$

在数字控制器的 *PID* 算法中，还有其他方面的改进，如输入滤波、梯形积分等。

4.1.3.2　类似 *PID* 控制规律的描述

采样控制系统中，在采样时刻之间，或在采样间隔之内，数字控制器的输出变化可以有不连续性。类似的其他控制规律如下：

①采样间断控制，在一个采样周期内，分成若干个时间片，每个时间片内有一定的控制规律。图 4－7 是几个典型的采样间断控制规律。

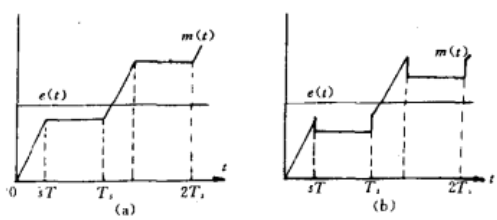


图 4－7　采样间断控制规律

图 4－7（*a*）是积分型采样间断控制规律。其子程序如下：

```
1000  DM (I) = KI * E (I)
1010  IF I > ST THEN DM (I) = KI * E (ST)
1020  M (I) = M (I) + DM (I)
1030  RETURN
```

图 4－7（*b*）是积分保持型二作用采样间断控制规律。其子程序如下：

```
1000  DM (I) = KC * (E (I) - E1 (I)) + KI * E (I)
1010  IF I > ST THEN DM (I) = KC * (E (ST) - E1 (ST))
1020  M (I) = M (I) + DM (I)
1030  E1 (I) = E (I)
1040  RETURN
```

程序中，DM（I）是增量输出，M（I）是控制器输出。I是在采样周期 T_s 内计算步数，$0<I<T_s$。ST是采样时间片的长度。KC、KI分别是数字控制器的比例与积分增益系数。E（I）、E1（I）分别是本次和上次的采样所得的偏差，E(I) = R(I) − Y(I)。R(I)是设定值。Y（I）是测量值（采样值）。

②预测 *PID* 控制：为了在 $k+1$ 拍预测 k 拍的偏差 $e(k)$，可假设 $e(k-1)$ 处的变化率不发生变化，即：

$$\frac{e(k)-e(k-1)}{T_s}=\frac{e(k-1)-e(k-2)}{T_s}$$

$$\therefore \quad e(k)=2e(k-1)-e(k-2) \tag{4-12}$$

由此，得预测 *PID* 控制算法的计算公式：

$$u(k)=K_1e(k-1)+K_2e(k-2)+K_3e^*(k-1) \tag{4-13}$$

式中，

$$K_1=2K_c+2\frac{K_c}{T_i}T_s+\frac{K_cT_d}{T_s}=2K_c+2K_I+K_D$$

$$K_2=-(K_c+K_I+K_D)$$

$$K_3=K_I$$

$$e^*(k-1)=e^*(k-2)+e(k-1)=\sum_{j=0}^{k-1}e(j)$$

（4－13）式改写成：

$$D(z)=\frac{U(z)}{E(z)}=K_c\left(\frac{2z-1}{z^2}\right)+K_I\frac{3z^2-3z+1}{z^2(z-1)}+K_D\frac{z-1}{z^2} \tag{4-14}$$

与滞后一拍的 *PID* 控制算法的计算公式

$$\frac{U(z)}{E(z)}=\frac{K_c}{z}+\frac{K_I}{z-1}+\frac{K_D(z-1)}{z^2} \tag{4-15}$$

相比较，可以看到预测 *PID* 控制算法中的比例与积分项引入了 $\frac{2z-1}{z}$ 及 $\frac{3z^2-3z+1}{z^2}$ 的环节。研究表明，由于引入了这些环节，提高了系统的稳定性。

作为 *PID* 控制算法的一般形式，可以写成：

$$D(z)=\frac{U(z)}{E(z)}=\frac{\beta_0+\beta_1z^{-1}+\cdots+\beta_pz^{-p}}{\alpha_0+\alpha_1z^{-1}+\cdots+\alpha_qz^{-q}}\qquad(\alpha\neq0) \tag{4-16}$$

满足稳态无余差的条件是控制器应有 $z=1$ 的极点，即：

$$D(z)=(\beta_0+\beta_1z^{-1}+\cdots+\beta_pz^{-p})/(1-z^{-1}) \tag{4-17}$$

或写成：

$$u(k)=u(k-1)+\beta_0e(k)+\beta_1e(k-1)+\cdots\beta_pe(k-p) \tag{4-18}$$

当 $p=0$ 得比例作用，$p=1$ 得比例积分作用，$p=2$ 得比例积分微分作用。

同样，根据（4－17），可写出预测 *PID* 控制算法的一般形式是：

$$\begin{aligned}u(k)=&u(k-1)+(2\beta_0+\beta_1)e(k-1)\\&+(\beta_2-1)e(k-2)+\beta_3e(k-3)+\cdots+\beta_pe(k-p)\end{aligned} \tag{4-19}$$

通过不同的控制稳定性判据的要求，确定相应的 β_0、β_1、β_2、$\cdots\beta_p$，从而得到不同的预测 *PID* 控制算法。

可以看到，式（4－19）很容易通过编程在计算机上实施。

4.2 采样控制系统的 z 域设计法

采样控制系统的设计常用的有 z 域设计法和时域设计法。这里仅介绍采用数字控制器的采样控制系统的设计方法。

如图 4－8 所示的采样控制系统，采用数字控制器的采样控制系统的设计是指根据所希望的系统闭环脉冲传递函数，如何来确定数字控制器的脉冲传递函数 $D(z)$。

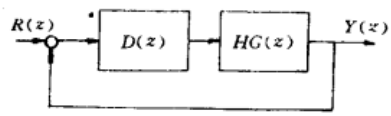


图 4－8 采样控制系统

假设所希望的闭环脉冲传递函数是 $\Phi(z)$。

$$\Phi(z) = \frac{Y(z)}{R(z)} = \frac{D(z)HG(z)}{1 + D(z)HG(z)} \tag{4-20}$$

式中，$HG(z) = Z[G_h(s)G_p(s)]$，是广义对象的脉冲传递函数。解（4－20）式，可得数字控制器 $D(z)$：

$$D(z) = \frac{1}{HG(z)} \cdot \frac{\Phi(z)}{1 - \Phi(z)} \tag{4-21}$$

若对象的 $HG(z)$ 已知，数字控制器的控制规律也已知，例如用上述的离散化的 PID 控制规律，则根据（4－20）可以求得该系统的闭环输出响应。若希望的闭环控制品质已知，即 $\Phi(z)$ 已知，则根据（4－21）可以求得数字控制器的控制规律。本节讨论数字控制器的 z 域设计法。

4.2.1 差拍控制算法

设 $HG(z)$ 含时滞项 z^{-d}，那么不管 $HG(z)$ 的输入是什么，在 dT_s 的时间内系统将不能响应（T_s 是采样周期），至少要在 $(d+1)$ 拍，即 $(d+1)T_s$ 时间才能响应。因此，$\Phi(z) = \frac{Y(z)}{R(z)}$ 必定含有时滞项，即 $C(z)$ 与 $R(z)$ 之间至少有 $(d+1)$ 拍的时滞，这种控制称为差拍控制。

控制系统设计时，首先应满足可实施的要求。即要求所设计的系统中各组成部件在物理上应能实现。

其次，静态无余差也是设计时应考虑的。这相当于连续系统中加入积分作用。如图 4－8 所示，偏差 $E(z)$

$$E(z) = R(z) - Y(z)$$

$$\because \quad \Phi(z) = \frac{Y(z)}{R(z)}$$

$$\therefore \quad E(z) = (1 - \Phi(z))R(z)$$

若静态无余差，则 $e(\infty) = 0$，应用终值定理，有

$$e(\infty) = \lim_{z \to 1}|(1 - z^{-1})E(z)| = \lim_{z \to 1}|(1 - z^{-1})[1 - \Phi(z)]R(z)| \tag{4-22}$$

若设定值 $R(z)$ 为阶跃、斜坡或加速度输入，则

$$R(z) = \frac{A(z)}{(1 - z^{-1})^m} \tag{4-23}$$

式中，$A(z)$ 是不含 $(1 - z^{-1})$ 的 z^{-1} 的多项式。将（4－23）代入（4－22），得：

$$1 - \Phi(z) = (1 - z^{-1})^m F(z) \tag{4-24}$$

式中，$F(z)$ 是分母项不含 $(1-z^{-1})$ 的待定的 z^{-1} 多项分式。(4－24) 式代入 (4－21) 式，得

$$D(z)=\frac{1}{HG(z)}\cdot\frac{1-(1-z^{-1})^{m}F(z)}{(1-z^{-1})^{m}F(z)}$$

这表明，为实现静态无余差，$D(z)$ 的分母项应含有 $(1-z^{-1})$ 项，它相当于连续系统中的积分环节 $\frac{1}{s}$。

根据对 $\Phi(z)$ 动态响应快慢的不同要求，或者对控制器输出变化的不同要求，可以设计出各种差拍控制系统。

4.2.1.1 最小拍控制算法

差拍控制系统中，若要求输出 $Y(z)$ 以最少的拍数跟踪设定值 $R(z)$ 的变化，即要求闭环脉冲传递函数 $\Phi(z)$ 中 z^{-1} 的次数应该最小。这种控制系统称最小拍控制系统。

根据 (4－24) 式，最小拍系统时 $F(z)=1$。

$$\therefore\quad (1-\Phi(z))=(1-z^{-1})^{m}$$

$$\Phi(z)=1-(1-z^{-1})^{m}$$

代入 (4－21) 式，得数字控制器的脉冲传递函数

$$D(z)=\frac{1}{HG(z)}\cdot\frac{1-(1-z^{-1})^{m}}{(1-z^{-1})^{m}}\qquad(4-25)$$

式中，m 的值需根据输入信号类型来确定。

设定值 $R(z)$ 若作阶跃变化，则 $R(z)=\frac{R}{1-z^{-1}}$。其中，R 是阶跃信号的幅值。根据无余差要求，由 (4－22) 式，有 $\lim\limits_{z\to1}[1-\Phi(z)]=0$

即在 $z=1$ 时，$\Phi(z)\big|_{z=1}=1$

代入 (4－21) 式，求得 $D(z)\big|_{z=1}=\infty$，这表明在 $z=1$ 处 $D(z)$ 有极点。因此，无法在物理上实施 $D(z)$，即 $C(z)$ 不可能在 $R(z)$ 变化的同一拍(采样周期)就跟踪上 $R(z)$ 的变化。根据(4－24)式，令 $m=1$，得 $1-\Phi(z)=1-z^{-1}$。即 $\Phi(z)=z^{-1}$。这表明闭环输出 $C(z)$ 滞后一拍才能响应 $R(z)$ 的变化。这时，求得

$$D(z)=\frac{1}{HG(z)}\cdot\frac{z^{-1}}{1-z^{-1}}\qquad(4-26)$$

它是可以实施的。(4－26) 式还表明，对于阶跃信号输入，要系统闭环输出跟踪上设定 $R(z)$ 的变化至少要滞后一拍。这是对于阶跃信号输入的最小拍数。采用同样的方法，可以求得，对于斜坡输入信号，最小拍数是两拍。对于加速度输入信号，最小拍数是三拍。

4.2.1.2 “跳动”现象

在最小拍控制系统中，存在两个问题。一个问题是“跳动”现象，另一个问题是控制器输出的能量极限。

为了使闭环输出以最小拍跟踪设定值 $R(z)$ 的变化，要求控制器输出足够大的能量，例如要使换热器的被加热物料出口温度从一个设定值变化到另一个设定值，为了最小拍跟踪，加热物料只能达到全开或全关，若这时提供的热量不足以在最小拍跟踪时，就会使闭环输出达不到所希望的指标。因此，在数字仿真时，必须要有输出值的限幅，或者采用以一定时间常数上升（或下降）来跟踪设定值 $R(z)$ 变化的大林算法或其他算法。

所谓“跳动”现象，又称振铃现象，它指数字控制器的输出有大幅度的上下摆动。产生跳动的原因可从脉冲传递函数来寻找。根据（4－20）式，$D(z)$ 与 $HG(z)$ 总是成对出现。若 $HG(z)$ 中出现极点，可以采用 $D(z)$ 中相应零点予以对消。但对于 $HG(z)$ 中出现的单位圆外零点和纯滞后 z^{-d}，则不允许用 $D(z)$ 的相应项对消。因为这会导致数字控制器有不稳定的极点和要求数字控制器有导前的输出。最小拍控制系统中，为了达到 $\Phi(z)$ 的要求，必然对于 $HG(z)$ 中的不稳定零点，用相应 $D(z)$ 的极点予以对消，这是造成跳动的根本原因。此外，由于 $HG(z)$ 不可能完全精确地反映对象的特性，当对象参数稍有变化，就会无法实现零极点的完全对消，这也会引入不稳定的输出。

减少跳动的一种办法是把 $HG(z)$ 中不稳定的零点包含在所设计的 $\Phi(z)$ 中，这将延长系统的调整时间。如V.E算法和卡尔曼算法。另一种方法是把 $D(z)$ 的具有跳动特性的极点项中的 z 用常数1代替，同时，对系统增益作相应调整。根据终值定理，当 $t\to\infty$ 时，$z\to1$，因此，这种替换方法不会影响最终稳态值。例如，$D(z)=\dfrac{5(1-0.7z^{-1})}{(1+0.8z^{-1})(1+z^{-1})(1-z^{-1})}$ 中的 $z=-0.8$ 与 $z=-1$ 是具有跳动特性的极点，用 $(1+0.8)$、$(1+1)$ 分别代替 $(1+0.8z^{-1})$ 及 $(1+z^{-1})$，这时，$D(z)$ 近似表示为：

$$D(z)=\frac{1.389(1-0.7z^{-1})}{(1-z^{-1})}$$

从实际系统来考虑，若系统的滞后项是 z^{-d}，则希望的 $\Phi(z)$ 至少滞后 $d+1$ 拍（对于阶跃信号输入），这时，$D(z)=\dfrac{1}{HG(z)}\cdot\dfrac{z^{-(d+1)}}{1-z^{-(d+1)}}$。当 $d\geqslant1$ 时，分母项 $(1-z^{-(d+1)})$ 总可以分解出 $(1+z^{-1})$ 的项，即 $D(z)$ 有 $z=-1$ 的极点，这会引起 $D(z)$ 输出的不衰减振荡。

4.2.1.3 示例

〔例4－3〕 分别对阶跃输入信号、斜坡输入信号及等加速度输入信号，列出最少拍系统的 $D(z)$。

①阶跃输入信号：$R(z)=\dfrac{R}{1-z^{-1}}$，R 是阶跃幅值。$\Phi(z)=z^{-1}$。

$$\therefore\qquad D(z)=\frac{1}{HG(z)}\cdot\frac{z^{-1}}{1-z^{-1}}$$

偏差 $E(z)=[1-\Phi(z)]\cdot R(z)=(1-z^{-1})\dfrac{R}{(1-z^{-1})}=R$

输出 $Y(z)$ 可由 $\dfrac{Y(z)}{E(z)}=D(z)\cdot HG(z)$ 求得

$$Y(z)=z^{-1}R(z)$$

或：

$$Y(k)=R(k-1)$$

输入与输出信号如图4－9所示。

②斜坡输入信号：$R(z)=\dfrac{VT_s z^{-1}}{(1-z^{-1})^2}$

对应的时域表示是 $R(t)=\begin{cases}Vt & t\geqslant0\\ 0 & t<0\end{cases}$

根据（4－22）式，有

$$\lim_{z\to 1}\left\{[1-\Phi(z)](1-z^{-1})\frac{VT_s z^{-1}}{(1-z^{-1})^2}\right\}=0 \quad \text{或} \quad \lim_{z\to 1}\frac{1-\Phi(t)}{1-z^{-1}}=0$$

为此，取 $[1-\Phi(z)]=(1-z^{-1})^2=1-2z^{-1}+z^{-2}$

$$\therefore \quad \Phi(z)=2z^{-1}-z^{-2}$$

$$\therefore \quad D(z)=\frac{1}{HG(z)}\cdot\frac{2z^{-1}-z^{-2}}{1-2z^{-1}+z^{-2}}$$

$$\therefore \quad \frac{Y(z)}{R(z)}=2z^{r1}-z^{-2}$$

或

$$y(k)=2R(k-1)-R(k-2)$$

偏差 $E(z)=VT_s z^{-1}$

以 $R(t)$ 代入得：

$$y(1)=2R(0)-R(-1)=2.0-0=0$$

$$y(2)=2R(1)-R(0)=2\cdot VT_s-0=2VT_s$$

$$y(3)=2R(2)-R(1)=2\cdot 2VT_s-VT_s=3VT_s$$

$$\cdots$$

$$y(n)=2\cdot R(n-1)-R(n-2)=2\cdot(n-1)VT_s-(n-2)VT_s=nVT_s \qquad (n\geqslant 3)$$

图 4－10 是斜坡输入时相应的输出值。

③等加速度输入信号：$R(t)=\begin{cases}\frac{A}{2}t^2 & t\geqslant 0\\ 0 & t<0\end{cases}$

其 z 变换是：$R(z)=\dfrac{AT_s^2 z^{-1}(1+z^{-1})}{2(1-z^{-1})^3}$

同样可求得 $\Phi(z)$ 满足 $1-\Phi(z)=(1-z^{-1})^3$ 时才能使静态无余差。$\therefore \Phi(z)=3z^{-1}-3z^{-2}+z^{-3}$

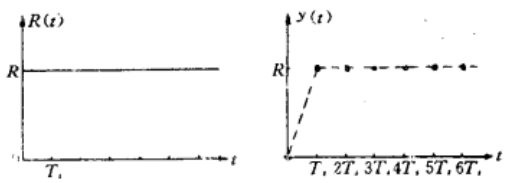


图 4－9　阶跃信号时输出响应

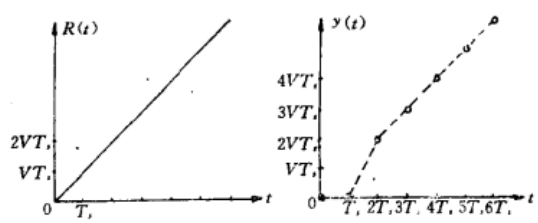


图 4－10　斜坡信号时输出响应

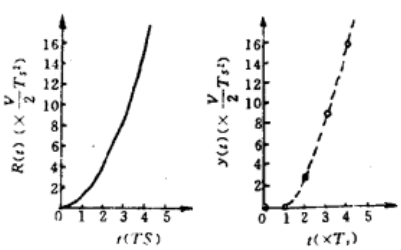


图 4－11　等加速度信号时输出响应

类似地求得：

$$E(z)=\frac{AT_s^2z^{-1}(1+z^{-1})}{2}$$

$$y(z)=(3z^{-1}-3z^{-2}+z^{-3})R(z)$$

或

$$y(k)=3R(k-1)-3R(k-2)+R(k-3)$$

图 4－11 是等加速度输入时相应的系统输出值。

应该指出，采用最小拍系统设计方法设计的数字控制器只适用于设计时特定的输入信号。

〔例 4－4〕　用斜坡信号输入时设计的数字控制器 $D(z)$，分别对阶跃信号、等加速度信号输入时计算系统的输出响应。

①阶跃输入信号：$R(z)=\dfrac{R}{1-z^{-1}}$

$$\because \quad D(z)=\frac{2z^{-1}-z^{-2}}{(1-z^{-1})^2}\cdot\frac{1}{HG(z)}$$

$$\therefore \quad \Phi(z)=\frac{D(z)HG(z)}{1+D(z)HG(z)}=\frac{\dfrac{2z^{-1}-z^{-2}}{(1-z^{-1})^2}}{1+\dfrac{2z^{-1}-z^{-2}}{(1-z^{-1})^2}}=2z^{-1}-z^{-2}$$

$$\therefore \quad Y(z)=(2z^{-1}-z^{-2})R(z)$$

或　　$$y(k)=2R(k-1)-R(k-2)$$

即 $y(0)=0;y(1)=2R;y(2)=2R-R=R;y(3)=R;\cdots$

系统输出响应如图 4－12 所示。可见，用高阶的输入信号设计的最小拍系统数字控制器 $D(z)$，用于低阶输入信号时，有较大的超调，同时调整时间加大。

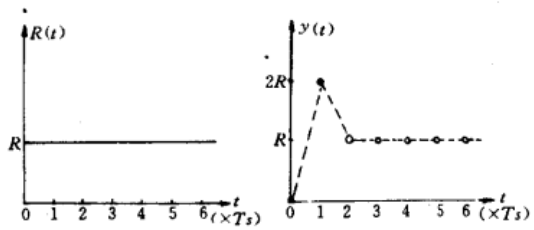


图 4－12　高阶 $D(z)$ 用于低阶输入信号

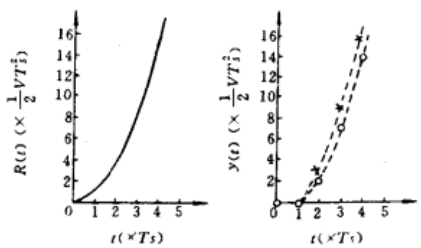


图 4－13　低阶 $D(z)$ 用于高阶输入信号

②等加速度输入信号：　$$R(z)=\frac{AT_s^2z^{-1}(1+z^{-1})}{2(1+z^{-1})^3}$$

$$\because \quad D(z)=\frac{2z^{-1}z^{-2}}{(1-z^{-1})^2}\cdot\frac{1}{HG(z)}$$

$$\therefore \quad \Phi(z)=2z^{-1}-z^{-2}$$

同样有：$Y(z)=(2z^{-1}-z^{-2})R(z)$

或 $y(k)=2R(k-1)-R(k-2)$

表 4－1 列出了相应的 $R(k)$ 与 $y(k)$。图 4－13 是相应的输出响应。可见，用低阶的输入信号设计的最小拍系统数字控制器 $D(z)$，用于高阶输入信号时，系统有稳态误差，不能很好跟踪 $R(z)$ 的变化。

表 4－1　系统输入、输出值

k	0	1	2	3	4	5	6
$R(k)$	0	$\frac{A}{2}T_s^2$	$2AT_s^2$	$\frac{9}{2}AT_s^2$	$8AT_s^2$	$\frac{25}{2}AT_s^2$	$18AT_s^2$
$y(k)$	0	0	AT_s^2	$\frac{7}{2}AT_s^2$	$7AT_s^2$	$\frac{23}{2}AT_s^2$	$17AT_s^2$

4.2.1.4　定值控制算式

上述算式是针对设定值变化而设计的。对于定值控制系统（如图 4－14 所示），它的设定值 $R(s)$ 等于零。因此，闭环输出 $Y(z)$ 由下式求出：

$$Y(z) = \frac{NG_p(z)}{1 + D(z)HG(z)} \tag{4-27}$$

因此，$D(z)$ 可由此求得

$$D(z) = \frac{NG_p(z) - Y(z)}{HG(z)Y(z)} \tag{4-28}$$

由于在设定值变化时，设定值变化的信息可以在采样时刻马上获得，而在干扰变化时，干扰进入回路的时间常常不是设计者所能确定。为此，在设计时，应按最坏情况进行考虑。若过程纯滞后为 dT_s，那么，在最坏的情况下，在 $dT_s + T_s$ 的采样时刻才能获得干扰变化的信息，并使控制器输出发生变化，由于过程的纯滞后，使闭环输出 $y(t)$ 要在 $2dT_s + T_s$ 时刻才能变化。这表明在最坏的情况下，系统接收到干扰的变化后，在 $dT_s + T_s$ 时间内，系统是处于开环状态的。在 $2dT_s + T_s$ 时间内，过程的输出没有反应，如图 4－15 所示。

这是在确定闭环脉冲传递函数时应该重视的问题。

［例 4－5］　对象 $G_p(s) = \frac{Ke^{-\tau s}}{Ts+1}$，取 T_s 使 $\tau = dT_s$。干扰为单位阶跃信号 $N(s) = \frac{1}{s}$，试确定数字控制器 $D(z)$。

$$HG(z) = \frac{K(1-\gamma)z^{-(d+1)}}{(1-\gamma z^{-1})} \quad ;\gamma = e^{-\frac{T_s}{T}}.$$

$$NG_p(z) = \frac{K(1-\gamma)z^{-(d+1)}}{(1-z^{-1})(1-\gamma z^{-1})}$$

根据（4－28）得：

$$D(z) = \frac{(1-\gamma)z^{-1} - (1-z^{-1})(1-\gamma z^{-1})\left[\sum_{i=1}^{d+1}(1-\gamma^i)z^{-i}\right]}{K(1-\gamma)(1-z^{-1})z^{-(d+1)}\left[\sum_{i=1}^{d+1}(1-\gamma^i)z^{-i}\right]}$$

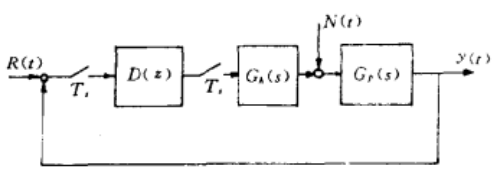


图 4－14　定值控制系统

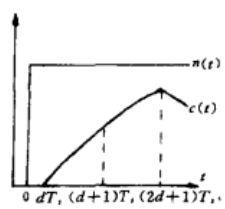


图 4－15　干扰输入时的响应

4.2.2　大林控制算法

4.2.2.1 大林控制算法的基本思想

差拍控制系统中，希望 $\Phi(z)=z^{(-d+1)}$，相当于连续系统的 $\frac{Y(s)}{R(s)}=e^{-\tau s}$，要求在 $d+1$ 拍起，一步达到设定值，从实际需要来说，似乎操之过急，而且实际系统不一定具有足够的能量使输出能一步到位，这就提出了大林(Dahlin)算法。

大林控制算法规定系统的输出 y（t）以一阶惯性环节的速率趋于设定值。即

$$\frac{Y(s)}{R(s)}=\frac{e^{-\tau s}}{\lambda s+1}$$

设 R（t）是阶跃函数 $R(t)=\begin{cases}1 & t\geqslant 0\\ 0 & t<0\end{cases}$，则 $R(s)=\frac{1}{s}$

$$\therefore \quad Y(s)=\frac{e^{-\tau s}}{\lambda s+1}\cdot\frac{1}{s}$$

化成离散形式，成为

$$Y(z)=\frac{(1-e^{-T_s/\lambda})z^{-(d+1)}}{(1-z^{-1})(1-e^{-T_s/\lambda}z^{-1})}=\frac{(1-\gamma)z^{-(d+1)}}{(1-z^{-1})(1-\gamma z^{-1})}$$

式中，$\gamma=e^{-\frac{T_s}{\lambda}}$，$T_s$ 是采样周期，$d=INT(\frac{\tau}{T_s})$

$$\because \quad R(z)=\frac{1}{(1-z^{-1})}$$

$$\therefore \quad \frac{Y(z)}{R(z)}=\frac{(1-\gamma)}{(1-\gamma z^{-1})}\cdot z^{-(d+1)} \qquad (4-29)$$

根据（4－21）式，得大林控制算法

$$D(z)=\frac{1}{HG(z)}\cdot\frac{(1-\gamma)z^{-(d+1)}}{1-\gamma z^{-1}-(1-\gamma)z^{-(d+1)}} \qquad (4-30)$$

当被控对象 $G_p(s)$ 是具有纯滞后的一阶或二阶非周期环节时，大林算法的计算公式如表4－2，表中的常数是：

$$b=e^{-\frac{T_s}{T}};c_1=1-e^{-m\frac{T_s}{T}};c_2=e^{-m\frac{T_s}{T}}-e^{-\frac{T_s}{T}};\alpha=e^{-\frac{T_s}{T_1}};\beta=e^{-\frac{T_s}{T_2}};$$

$$\gamma=e^{-\frac{T_s}{\lambda}};c_3=1+\frac{T_1\alpha-T_2\beta}{T_2-T_1};c_4=\alpha\beta+\frac{T_1\beta-T_2\alpha}{T_2-T_1} \qquad (4-31)$$

表4－2 大林控制算式

$G_p(s)$	τ 与 T_s 关系	$HG(z)=Z\left\{\frac{1-e^{-T_s s}}{s}\cdot G_p(s)\right\}$	$D(z)=\frac{1}{HG(z)}\cdot\frac{\Phi(z)}{1-\Phi(z)}$
$\frac{Ke^{-\tau s}}{Ts+1}$	$\tau=dT_s$	$\frac{K(1-b)z^{-(d+1)}}{(1-bz^{-1})}$	$\frac{(1-\gamma)(1-bz^{-1})}{K[1-\gamma z^{-1}-(1-\gamma)z^{-(d+1)}](1-b)}$
$\frac{Ke^{-\tau s}}{Ts+1}$	$\tau=dT_s+\Delta$ $m=1-\frac{\Delta}{T_s}$	$\frac{K(c_1+c_2z^{-1})z^{-(d+1)}}{(1-bz^{-1})}$	$\frac{(1-\gamma)(1-bz^{-1})}{K(c_1+c_2z^{-1})[1-\gamma z^{-1}-(1-\gamma)z^{-(d+1)}]}$
$\frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}$	$\tau=dT_s$	$\frac{K(c_3+c_4z^{-1})z^{-(d+1)}}{(1-\alpha z^{-1})(1-\beta z^{-1})}$	$\frac{(1-\gamma)(1-\alpha z^{-1})(1-\beta z^{-1})}{K(c_3+c_4z^{-1})[1-\gamma z^{-1}-(1-\gamma)z^{-(d+1)}]}$

［例4－6］ 已知对象 $G_p(s)=\frac{e^{-0.76s}}{0.4s+1}$，取 $T_s=0.5$ 求大林算法的 $D(z)$。

取 $\lambda=0.15$，$\because d=1$，$\therefore \theta=0.76-0.5=0.26$

$$\therefore \quad HG(z)=(1-z^{-1})Z\left\{\frac{e^{-0.76s}}{s(0.4s+1)}\right\}=(1-z^{-1})Z\left\{\frac{e^{-0.26s}e^{-0.5s}}{s(0.4s+1)}\right\}$$

$$= (1-z^{-1})z^{-1}Z\left\{\frac{e^{-0.26s}}{s(0.4s+1)}\right\} = (1-z^{-1})z^{-1}Z_m\left\{\frac{1}{s(0.4s+1)}\right\}$$

$$\because \quad z_m\left\{\frac{a}{s(s+a)}\right\} = z^{-1}\left\{\frac{1}{1-z^{-1}} - \frac{e^{-amT_s}}{1-e^{-aT_s z^{-1}}}\right\}$$

令
$$a = \frac{1}{0.4} = 2.5, m = 1 - \frac{\theta}{T_s} = 0.48$$

$$\therefore \quad Z_m\left\{\frac{1}{s(0.4s+1)}\right\} = Z_m\left\{\frac{2.5}{s(s+2.5)}\right\} = Z^{-1}\left\{\frac{1}{1-z^{-1}} - \frac{0.5488}{1-0.2865z^{-1}}\right\}$$

$$= \frac{z^{-1}(0.4512+0.2623z^{-1})}{(1-z^{-1})(1-0.2865z^{-1})}$$

$$\therefore \quad HG(z) = \frac{0.4512z^{-2}+0.2623z^{-3}}{1-0.2865z^{-1}}$$

$$\therefore \quad D(z) = \frac{1}{HG(z)} \cdot \frac{(1-e^{-\frac{T_s}{\lambda}})z^{-2}}{1-e^{-\frac{T_s}{\lambda}}z^{-1}-(1-e^{-\frac{T_s}{\lambda}})z^{-2}}$$

$$= \frac{0.964-0.23z^{-1}}{0.45+0.242z^{-1}-0.443z^{-2}-0.251z^{-3}}$$

$$= \frac{2.142-0.622z^{-1}}{1+0.542z^{-1}-0.984z^{-2}-0.558z^{-3}}$$

$$\because \quad D(z) = \frac{M(z)}{E(z)}$$

$$\therefore \quad M(k) = 2.142E(k) - 0.622E(k-1) - 0.542M(k-1) + 0.984M(k-2) + 0.558M(k-3)$$

数字仿真时，应选择合适的 λ 值。方法是先设置一个初值，然后通过数字仿真观察输出响应，通过反复修改 λ 和观察输出响应，确定合适的 λ 值。

4.2.2.2 改进大林控制算法

只从 $y(t)$ 的响应来评价控制算法的性能是不够的，图 4－16 是采用大林算法时数字控制器的输出响应，可以看到“跳动”现象也存在于大林控制算法中。改进大林控制算法就是采用常数 1 代替 $D(z)$ 中具有跳动特性的极点中的 z 项来减少跳动的。图 4－17 是改进后数字控制器的输出响应。可以看到，系统的输出与原来的相似，而跳动明显改善。

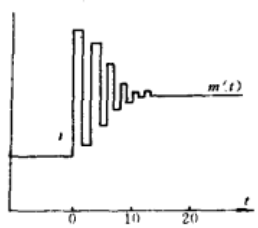


图 4－16 大林算法的 $D(z)$ 输出

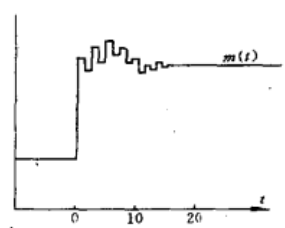


图 4－17 改进大林算法的 $D(z)$ 输出

4.2.3 卡尔曼控制算法

4.2.3.1 卡尔曼控制算法

由上述可知，只对 $Y(z)$ 或 $\frac{Y(z)}{R(z)}$ 规定所需的特性，不一定能得到理想的效果，为了保

证数字控制器的输出不出现“跳动”现象，卡尔曼（Kalman）控制算法假设 $\frac{M(z)}{R(z)}=Q(z)$，$Q(z)$ 是具有稳定极点的脉冲传递函数。同时，假设 $\frac{Y(z)}{R(z)}=P(z)$。$P(z)$ 也是具有稳定极点的脉冲传递函数。式中，$M(z)$ 是数字控制器输出的脉冲传递函数。

因此，可得

$$\frac{Y(z)}{M(z)}=HG(z)=\frac{P(z)}{Q(z)} \tag{4-32}$$

根据（4-21），即得

$$D(z)=\frac{Q(z)}{P(z)}\cdot\frac{P(z)}{1-P(z)}=\frac{Q(z)}{1-P(z)} \tag{4-33}$$

当 $1-P(z)$ 的特征根位于单位圆内时，就能消除“跳动”现象。

可以看到，本算法对于随动系统有：

$$\left.\frac{Y(z)}{R(z)}\right|_{z=1}=P(z)|_{z=1}=1 \tag{4-34}$$

对于定值系统，有：

$$\left.\frac{Y(z)}{N(z)}\right|_{z=1}=1-P(z)|_{z=1}=0$$

根据 $\frac{M(z)}{R(z)}=Q(z)$，可以得到：

$$\left.\frac{Y(z)}{M(z)}\right|_{z=1}=\frac{1}{Q(z)|_{z=1}}=K \tag{4-35}$$

式中，K 是广义对象的增益。

表 4-3 列出了卡尔曼控制算法的计算公式，表中的常数定义见（4-31）式。

表 4-3 卡尔曼控制算式

$G_p(s)$	τ 与 T_s 关系	$HG(z)$	$P(z)$
$\frac{Ke^{-\tau s}}{Ts+1}$	$\tau=dT_s$	$\frac{K(1-b)z^{-(d+1)}}{(1-bz^{-1})}$	$z^{-(d+1)}$
$\frac{Ke^{-\tau s}}{Ts+1}$	$\tau=dT_s+\Delta$ $m=1-\frac{\Delta}{T_s}$	$\frac{K(c_1+c_2z^{-1})z^{-(d+1)}}{(1-bz^{-1})}$	$\frac{(c_1+c_2z^{-1})}{c_1+c_2}z^{-(d+1)}$
$\frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}$	$\tau=dT_s$	$\frac{K(c_3+c_4z^{-1})z^{-(d+1)}}{(1-\alpha z^{-1})(1-\beta z^{-1})}$	$\frac{(c_3+c_4z^{-1})}{c_3+c_4}\cdot Z^{-(d+1)}$

$G_p(s)$	$Q(z)$	$D(z)=\frac{Q(z)}{1-P(z)}$
$\frac{Ke^{-\tau s}}{Ts+1}$	$\frac{(1-bz^{-1})}{K(1-b)}$	$\frac{(1-bz^{-1})}{K(1-b)(1-z^{-(d+1)})}$
$\frac{Ke^{-\tau s}}{Ts+1}$	$\frac{(1-bz^{-1})}{K(c_1+c_2)}$	$\frac{(1-bz^{-1})}{K(c_1+c_2)\left[1-\frac{c_1+c_2z^{-1}}{c_1+c_2}z^{-(d+1)}\right]}$
$\frac{Ke^{-\tau s}}{(T_1s+1)(T_2s+1)}$	$\frac{(1-\alpha z^{-1})(1-\beta z^{-1})}{K(c_3+c_4)}$	$\frac{(1-\alpha z^{-1})(1-\beta z^{-1})}{K(c_3+c_4)\left[1-\frac{c_3+c_4z^{-1}}{c_3+c_4}z^{-(d+1)}\right]}$

可以看到，表4－3中的$P(z)$与$Q(z)$是满足（4－34）与（4－35）式的。这表明采用卡尔曼控制算法，既能跟踪设定值$R(z)$的变化，而无余差，又能克服扰动的影响。

4.2.3.2 示例

［例4－7］　已知$G_p(s)=\dfrac{e^{-2s}}{(s+1)(0.8s+1)}$，取$T_s=1$，计算卡尔曼控制算式。

根据表4－3，直接求得：

$$D(z)=\frac{Q(z)}{1-P(z)}=\frac{(1-0.2865z^{-1})(1-0.3678z^{-1})}{(0.3+0.144)\left[1-\dfrac{0.3+0.144z^{-1}}{0.3+0.144}z^{-3}\right]}$$

$$=\frac{2.252(1-0.6543z^{-1}+0.1054z^{-2})}{(1-0.6757z^{-3}-0.3243z^{-4})}$$

写成递推算式是：

$$M(k)=0.6757M(k-3)+0.3243M(k-4)$$
$$+2.252[E(k)-0.6543E(k-1)+0.1054E(k-2)]$$

〔例4－8〕　已知$G_p(s)=\dfrac{e^{-1.46s}}{3.3s+1}$，取$T_s=1$，计算卡尔曼控制算式。

根据表4－3，得：$m=0.54$

$$D(z)=\frac{1-bz^{-1}}{K(c_1+c_2)\left[\left(1-\dfrac{c_1+c_2z^{-1}}{c_1+c_2}\right)z^{-2}\right]}$$

$$=\frac{3.826(1-0.7386z^{-1})}{1-0.5773z^{-2}-0.4324z^{-3}}$$

控制器的输出可由下式递推计算：

$M(k)=0.5773M(k-2)+0.4243M(k-3)+3.826[E(k)-0.7386E(k-1)]$

可以看到，卡尔曼控制算法使闭环脉冲传递函数成为$P(z)$，它不是最小拍，如本例中

$$P(z)=\frac{c_1+c_2z^{-1}}{c_1+c_2}z^{-2}=(0.5773+0.4243z^{-1})z^{-2}$$

这表明闭环输出响应中含有$z=-0.735$的零点。

4.2.4　V.E. 控制算法

4.2.4.1　V.E. 控制算法

如前所述，当$HG(z)$含有不稳定的零点时，根据（4－21）式，$D(z)$将含有不稳定的极点，这些极点造成数字控制器$D(z)$输出的剧烈波动。虽然，在各采样时刻系统的输出符合性能要求，但在采样间隔系统输出的波动还是造成了系统的不稳定，沃格尔和埃德加（*Vogel*，*E*．*F*．*&Edgar*，*T*．*F*．）提出的*V*．*E*．控制算法是假设$HG(z)$可分解为稳定和不稳定两部分。即

$$HG(z)=\frac{B_1(z)}{A_1(z)}\cdot\frac{B_2(z)}{A_2(z)}\cdot z^{-(d+1)}\qquad(4-36)$$

式中，$B_1(z)$、$A_1(z)$为不稳定部分，$B_2(z)$、$A_2(z)$为稳定部分。V.E. 控制算法假设$D(z)$只对消$HG(z)$中的一部分，而保留$HG(z)$中的不稳定零点，即

$$D(z)=\frac{A_1(z)A_2(z)}{B_2(z)}P_B(z)\qquad(4-37)$$

根据大林算法的要求，

$$\frac{Y(z)}{R(z)} = \frac{(1-\gamma)z^{-(d+1)}}{(1-\gamma)z^{-1}}$$

由于 $HG(z)$ 中不稳定零点未对消，因此，$V.E.$ 控制算法中，

$$\frac{Y(z)}{R(z)} = \frac{K_B(1-\gamma)B_1(z)}{(1-\gamma z^{-1})}z^{-(d+1)} \tag{4-38}$$

根据稳态条件，$\lim_{t\to\infty}\frac{y(t)}{R(t)}=1$，得

$$K_B = \frac{1}{B_1(z)\mid_{z=1}} = \frac{1}{B_1(1)}$$

根据闭环脉冲传递函数的计算公式：

$$\frac{Y(z)}{R(z)} = \frac{D(z)HG(z)}{1+D(z)HG(z)}$$

对（4－36）、（4－37）、（4－38）联立求解，得：

$$P_B(z) = \frac{K_B(1-\gamma)}{\left[(1-\gamma z^{-1}) - B_1(z)\frac{1}{B_1(1)}(1-\gamma)z^{-(d+1)}\right]}$$

$$\therefore D(z) = \frac{A_1(z)A_2(z)}{B_2(z)} \cdot \frac{(1-\gamma)}{[B_1(1)(1-\gamma z^{-1}) - B_1(z)(1-\gamma)z^{-(d+1)}]} \tag{4-39}$$

（4－39）式表明，当〔$B_1(1)(1-\gamma z^{-1}) - B_1(z)(1-\gamma)z^{-(d+1)}$〕的特征根是稳定的，数字控制器的输出就没有“跳动”现象。

4.2.4.2 卡尔曼控制算法与 $V.E.$ 控制算法的关系

在卡尔曼控制算法中，假设 $HG(z) = \frac{B_1(z)B_2(z)}{A_1(z)A_2(z)} \cdot z^{-(d+1)}$，与 $V.E.$ 控制算法的假设一致，则有

$$\frac{y(z)}{R(z)} = \frac{y(z)}{M(z)} \cdot \frac{M(z)}{R(z)} = HG(z) \cdot Q(z) = \frac{B_1(z)B_2(z)}{A_1(z)A_2(z)} \cdot z^{-(d+1)} \cdot Q(z) \tag{4-40}$$

$V.E.$ 控制算法中，有：

$$\frac{Y(z)}{R(z)} = \frac{K_B(1-\gamma)B_1(z)}{(1-\gamma z^{-1})} \cdot z^{-(d+1)} \tag{4-41}$$

（4－41）与（4－40）相等，有：

$$Q(z) = \frac{A_1(z)A_2(z)}{B_2(z)} \cdot \frac{K_B(1-\gamma)}{(1-\gamma z^{-1})} \tag{4-42}$$

可以看到

$$P(z) = \frac{Y(z)}{R(z)} = \frac{K_B(1-\gamma)B_1(z)}{(1-\gamma z^{-1})} \cdot z^{-(d+1)}$$

满足下式：

$$P(z)\mid_{z=1} = 1$$

同样：

$$Q(z) = \frac{A_1(z)A_2(z)}{B_2(z)} \cdot \frac{K_B(1-\gamma)}{(1-\gamma z^{-1})}$$

满足下式：

$$Q(z)\mid_{z=1} = \frac{1}{K}$$

式中，K 是对象增益。

因此，当选用 $P(z)$、$Q(z)$ 满足（4－41）及（4－42）时，卡尔曼控制算法与 V.E. 控制算法是完全一致的。

根据（4－36）与（4－39）式，可以看到，当 $B_1(z) = 1$ 时，即 $HG(z)$ 不含有不稳定零点时，V.E. 控制算法与大林算法是一致的。

［例 4－9］　已知对象 $G_p(s) = \dfrac{e^{-s}}{(3s+1)(5s+1)}$，取 $T_s = 1$。由于

$$HG(z) = \frac{0.02797 + 0.0234z^{-1}}{1 - 1.5353z^{-1} + 0.5866z^{-2}} \cdot z^{-2}$$

得出存在不稳定零点的结果。为此，采用 $\lambda = 1$ 的 V.E. 控制算法。

$$D(z) = \frac{A_1(z)A_2(z)}{B_2(z)} \cdot \frac{1-\gamma}{B_1(1)(1-\gamma z^{-1}) - B_1(z)(1-\gamma)z^{-(d+1)}}$$

取：

$$B_1(z) = 0.02797 + 0.0234z^{-1}; B_2(z) = 1;$$

$$A_1(z) \cdot A_2(z) = (1 - 1.5353z^{-1} + 0.5866z^{-2}); d = 1;$$

得：

$$D(z) = \frac{12.3(1 - 1.5353z^{-1} + 0.5866z^{-2})}{1 - 0.368z^{-1} - 0.344z^{-2} - 0.288z^{-3}}$$

可以看到，跳动现象明显得到改善。

4.3　含有纯滞后的采样控制系统

纯滞后对象的采样控制系统除了可采用上节所述的 z 域设计法设计控制器外，还可采用纯滞后补偿的控制方案。纯滞后 $e^{-\tau s}$ 的实施采用计算机要比常规仪表来得方便，因此，纯滞后补偿的采样控制系统得到了广泛应用，为了方便分析和讨论，本节在分析控制方案时采用了 $e^{-\tau s}$ 表示纯滞后，在实施时则采用它的时间离散形式。考虑到采样系统，还需加入由采样引入的纯滞后的影响。

4.3.1　史密斯预估补偿器

由史密斯（O. J. M. Smith）在 1957 年提出的以模型为基础的补偿纯滞后影响的方法由于缺乏实施的实际硬件，因此，虽然这种算法早已提出，但一直未能广泛应用。数字计算机的出现使史密斯预估补偿器获得新的生命，并得到了许多应用的验证。

4.3.1.1　史密斯预估补偿器的基本原理

图 4－18 是含有纯滞后环节的对象，采用常规控制时的控制系统框图。可以看到，对象 $G_p(s)e^{-\tau_p s}$ 可分解为两部分，一部分不含纯滞后，用 $G_p(s)$ 表示，另一部分是纯滞后部分

$e^{-\tau_p s}$。

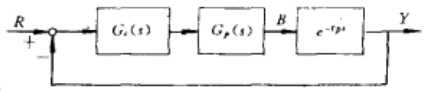


图 4－18　含有纯滞后的控制系统

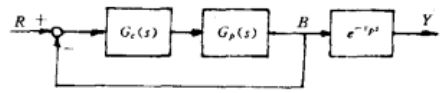


图 4－19　反馈回路的理想结构

如果能设法把假想的变量 B 测量出来，如图 4－19 所示，把 B 信号输到控制器，就能把纯滞后环节移到控制回路之外，而输出 Y 经 τ_p 的延时后将重复 B 信号的变化。与图 4－18 相比较，图 4－19 的系统闭环特征方程中不含有纯滞后项，因此有可能提高控制器增益使系统响应大大改善。因为 B 信号是不可测的假想信号，因此在实施时，引入史密斯预估补偿器 $G_k(s)$，如图 4－20 所示。

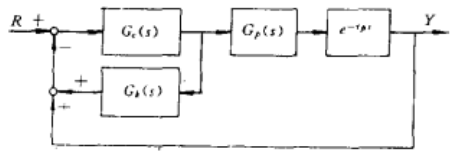


图 4－20　史密斯预估补偿控制系统

为使图 4－20 与图 4－19 的输出 $Y(s)$ 与设定 $R(s)$ 的传递函数等效，应有：

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)e^{-\tau_p s}}{1 + G_c(s)G_p(s)} = \frac{G_c(s)G_p(s)e^{-\tau_p s}}{1 + G_c(s)G_k(s) + G_c(s)G_p(s)e^{-\tau_p s}} \qquad (4-43)$$

即有：

$$G_k(s) = G_p(s)(1 - e^{-\tau_p s}) \qquad (4-44)$$

这表明，若采用 $G_k(s)$ 如（4－44）所示的传递函数，就能使反馈信号相当于从 B 处引出，即提前了 τ_p 时刻，所以这种控制称为预估补偿控制。

4.3.1.2　史密斯预估补偿器的实施

史密斯预估补偿控制系统中，要求补偿器 $G_k(s)$ 具有下列传递函数。

$$G_k(s) = G_p(s)(1 - e^{-\tau_p s})$$

由于对象特性不能精确获得，在具体实施时，采用图 4－21 的方案。图中，$G_m(s)(1 - e^{-\tau_m s})$ 代替 $G_k(s)$，因此，闭环传递函数是

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G_p(s)e^{-\tau_p s}}{1 + G_c(s)G_m(s) - G_c(s)G_m(s)e^{-\tau_m s} + G_c(s)G_p(s)e^{-\tau_p s}} \qquad (4-45)$$

闭环特征方程是

$$1 + G_c(s)G_m(s) - G_c(s)G_m(s)e^{-\tau_m s} + G_c(s)G_p(s)e^{-\tau_p s} = 0 \qquad (4-46)$$

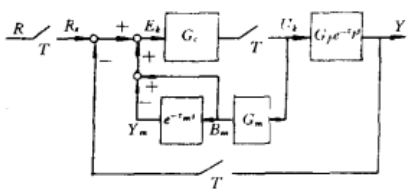


图 4－21　史密斯预估补偿控制系统的实施

当 $G_m(s) = G_p(s)$ 及 $\tau_m = \tau_p$ 时，闭环特征方程成为

$$1 + G_c(s)G_p(s) = 0$$

这时，（4－45）式与（4－43）相同，构成了以假想变量 B 为反馈信号的理想结构时的闭环传递函数。

因此，实施史密斯预估补偿控制方案要解决两个问题。即模型的辨识及纯滞后环节的实现。

〔例 4－10〕　已知 $G_p(s)=\dfrac{K_p}{T_ps+1}$，纯滞后为 τ_p 的对象，采用史密斯预估补偿控制方案，确定史密斯预估补偿器的输入和输出的算式。

如图 4－21 所示，模型输出 Y_m 与输入 U 之间的关系可用下式表示：

$$\frac{Y_m(z)}{U(z)}=Z\{G_{h0}G_m(s)e^{-\tau_m s}\} \tag{4-47}$$

式中，$Y_m(z)$ 是模型输出 Y_m 的 z 变换。$U(z)$ 是输入的 z 变换。$G_{h0}(s)$ 是零阶保持器的传递函数，$G_{h0}(s)=\dfrac{1-e^{-sT_s}}{s}$。$G_m(s)$ 是模型的传递函数，$G_m(s)=\dfrac{K_m}{T_ms+1}$。$e^{-\tau_m s}$ 是模型的纯滞后项，τ_m 是纯滞后。

设纯滞后 τ_m 与采样周期 T_s 之间有：

$$\tau_m=(N+\beta)T_s \tag{4-48}$$

N 是 τ_m 中所含 T_s 个数的整数部分，$N=\mathrm{INT}\left(\dfrac{\tau_m}{T_s}\right)$。而 β 是 0 ～ 1 之间的小数，$\beta=\dfrac{\tau_m}{T_s}-\mathrm{INT}\left(\dfrac{\tau_m}{T_s}\right)$。

(4－47) 代入 (4－48)，取 z 变换得：

$$\frac{Y_m(z)}{U(z)}=K_mz^{-N}(1-z^{-1})z^{-1}\left(\frac{1}{1-z^{-1}}-\frac{e^{-(1-\beta)T_i/T_m}}{1-e^{-T/T_m}z^{-1}}\right) \tag{4-49}$$

交叉相乘，并进行反变换，得到递推算式：

$$Y_{m,k}=A_2Y_{m,(k-1)}+A_2K_m\left(\frac{1}{A_3}-1\right)U_{k-(N+2)}+K_m\left(1-\frac{A_2}{A_3}\right)U_{k-(N+1)} \tag{4-50}$$

式中，$A_2=e^{-\frac{T_s}{T_m}}$

$A_3=e^{-\beta\frac{T_s}{T_m}}$

同样，可以求出不含纯滞后的输出 $B_m(z)$ 与 $U(z)$ 的关系是：

$$\frac{B_m(z)}{U(z)}=K_m(1-z^{-1})\left(\frac{1}{1-z^{-1}}-\frac{1}{1-e^{-\frac{T}{T_m}}z^{-1}}\right) \tag{4-51}$$

并求得递推算式：

$$B_{m,k}=K_m(1-A_2)U_{k-1}+A_2B_{m,(k-1)} \tag{4-52}$$

控制器输入端的偏差信号按下式计算：

$$E_{k+1}=R_k-Y_k-(B_{m,k}-Y_{m,k}) \tag{4-53}$$

可以看到，Y_k 与 U_k 的关系与 (4－50) 相似，是

$$Y_k=A_4Y_{k-1}+A_4K_p\left(\frac{1}{A_5}-1\right)U_{k-(N+2)}+K_p\left(1-\frac{A_4}{A_5}\right)U_{k-(N+1)} \tag{4-54}$$

式中，$A_4=e^{-\frac{T_s}{T_p}}$

$A_5=e^{-\beta\frac{T_s}{T_p}}$

当 K_p、T_p、τ_p 与模型的 K_m、T_m、τ_m 对应相等时，有：

$$Y_k=Y_{m,k}$$

因此，控制器的输入信号成为

$$E_{k+1} = R_k - B_{m,k} \tag{4-55}$$

过程所包含的纯滞后被补偿。这时的控制器可以采用常规的控制算法。

4.3.2 观测补偿器

4.3.2.1 基本原理和结构

根据现代控制理论，对于一个确定性的系统，可以采用观测器的方法，通过对观测值与输出值之间的误差进行闭环校正，来间接获得系统的状态量。

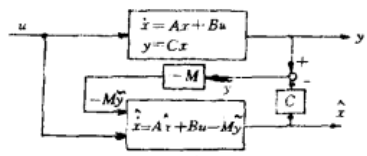


图 4－22 状态观测器方框图

图 4－22 是状态观测器方框图。设控制系统是

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{4-56}$$
$$y(t) = Cx(t)$$

为获得状态量 $x(t)$，构造一个原系统的模型：

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t)$$
$$\hat{y}(t) = C\hat{x}(t)$$

考虑到模型的不准确和量测噪声的影响，加入闭环校正项，使模型成为：

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) - M[y(t) - \hat{y}(t)] \\ &= A\hat{x}(t) + Bu(t) - M\tilde{y}(t)\end{aligned} \tag{4-57}$$

令 $\tilde{x}(t) = x(t) - \hat{x}(t)$，则有

$$\dot{\tilde{x}}(t) = \dot{x}(t) - \dot{\hat{x}}(t) = A[x(t) - \hat{x}(t)] + M\tilde{y}(t) = [A + MC]\tilde{x}(t) \tag{4-58}$$

当 $\hat{x}(0) = x(0)$ 已知时，求解上述方程可得：

$$\tilde{x}(t) = e^{(A+MC)t}\tilde{x}(0) \tag{4-59}$$

这表明，状态误差向量与输入量无关。当系统稳定条件满足时，$\tilde{x}(t)$ 趋于零，即 $\hat{x}(t) = x(t)$。而系统的稳定性通过 M 的选择来满足。

当系统是时变或非线性时，或者由于量测噪声，原系统的模型不可能完全与实际系统一致，为使闭环校正后，状态误差向量为零，应在校正项中加入积分环节。组成图 4－23 所示的系统。加入增广向量 $q(t)$：

$q(t) = \int_0^t [y(t) - \hat{y}(t)]dt$，得其导数项 $\dot{q}(t)$：

$$\dot{q}(t) = y(t) - \hat{y}(t) = y(t) - C\hat{x}(t)$$

因此，系统增广为：

$$\begin{bmatrix}\dot{\hat{x}}(t) \\ \dot{q}(t)\end{bmatrix} = \begin{bmatrix}A + BK_1 & BK_2 \\ -C & 0\end{bmatrix}\begin{bmatrix}\hat{x}(t) \\ q(t)\end{bmatrix} + \begin{bmatrix}Bu(t) \\ y(t)\end{bmatrix}$$

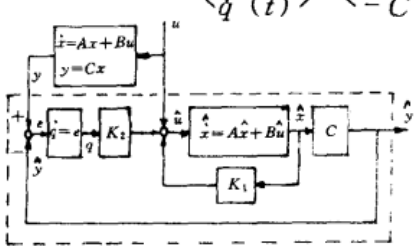


图 4－23 具有积分环节的观测器

$$\hat{y}(t) = [C \quad 0]\begin{bmatrix}\hat{x}(t) \\ q(t)\end{bmatrix} \tag{4-60}$$

通过增益矩阵 $[K_1 \quad K_2]$ 的选择，可使增广系统闭环极点任意配置。例如，若选 $K = [0 \quad K_2]$，则闭环系统能稳定，且使 $\hat{x}(t)$、$q(t)$ 对于阶跃输入

的 $u(t)$ 和 $y(t)$ 均能收敛于稳态，即满足 $\lim_{t\to\infty}\hat{y}(t)=y(t)$。由于 $\hat{y}(t)=C\hat{x}(t)$ 及 $y(t)=Cx(t)$，因此有 $\hat{x}(t)=x(t)$。这表明，加入积分环节后，系统的估值和实际值无差。

对于纯滞后对象，采用计算机控制的观测补偿器控制系统如图 4－24 所示。闭环特征方程是

$$1+G_c(s)G_m(s)\frac{1+G_k(s)G_0(s)}{1+G_k(s)G_m(s)}=0 \qquad (4-61)$$

图中，$G_m(s)$ 相当于观测器，$G_k(s)$ 是闭环输出的校正环节，采用具有积分作用的控制算法。若满足 $G_k(s)$ 的模足够小，就有：

$$1+G_k(s)G_0(s)\doteq 1$$

$$1+G_k(s)G_m(s)\doteq 1$$

这时，闭环特征方程成为

$$1+G_c(s)G_m(s)=0$$

这与史密斯预估补偿器控制时的结果相同。可以看到，史密斯预估补偿控制方案采用在闭环特征方程中含有纯滞后的项相互加减来消除纯滞后项，观测器补偿控制方案则采用相除的方法使纯滞后项作用减小。

本方案的调节过程说明如下。对于设定值变化的随动控制系统，本方案可看成由两个随动控制系统组成。由主控制器 $G_c(s)$ 与 $G_0(s)$、$G_m(s)$、$G_k(s)$ 组成主随动控制系统，用于使观测器输出值跟踪设定值。由副控制器 $G_k(s)$ 与 $G_m(s)$ 组成的副随动控制系统，用于使观测器输出值与系统输出值保持同步。当设定值 R 变化时，由于观测器输出 Y_m 尚未变化，所以，主控制器输出一个较强的调节信号，通过前馈作用，使观测器输出较快地跟踪，减小控制器的偏差，起到超前调节的作用。与此同时，该调节信号对对象本身也起调节作用，使输出 Y 较快变化。而副控制器的调节作用，保证了观测器输出 Y_m 与系统输出 Y 间的跟踪。当选用含有积分作用的主、副控制器时，可使系统稳态无余差。

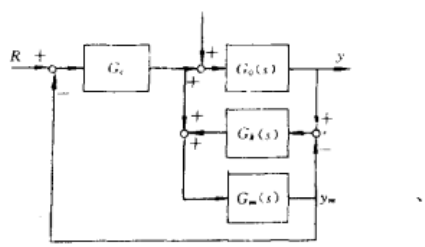


图 4－24　观测补偿器控制方案

4.3.2.2　观测补偿控制方案的实施

采用计算机控制时，观测补偿器控制方案可采用图 4－25 所示的一般结构形式。当图中的 τ_m 取零值时，等效于图 4－24。

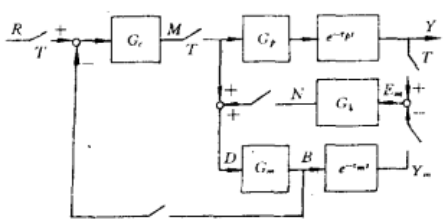


图 4－25　观测补偿器控制方案的实施

〔例 4－11〕　已知对象 $G_p(s)=\dfrac{K_p}{T_p s+1}$；$G_0=G_p(s)e^{-\tau_p s}$。采用模型 $G_m(s)=\dfrac{K_m}{T_m s+1}$，取 $\tau_m=0$。列写观测器输入与输出关系。

如图 4－25 所示。观测器输入由两部分相加：

$$D_k=M_k+N_k \qquad (4-62)$$

由于 $G_m(s)=\dfrac{K_m}{T_m s+1}$，经零阶保持器后，可得

$$\frac{B(z)}{D(z)} = K_m(1-z^{-1})\left[\frac{1}{1-z^{-1}} - \frac{1}{1-e^{-\frac{T}{T_m}}z^{-1}}\right] \tag{4-63}$$

即：

$$B_k = K_m(1-A_2)U_{k-1} + A_2B_{k-1} \tag{4-64}$$

式中，$A_2 = e^{-\frac{T}{T_m}}$

由于 $\tau_m = 0$，因此，$Y_{m,k} = B_k$

副控制器输入是系统输出 Y 与观测器输出 Y_m 之差，即有

$$E_{m,k+1} = Y_k - Y_{m,k} \tag{4-65}$$

同样，主控制器输入的偏差 E_{k+1} 是：

$$E_{k+1} = R_k - B_k \tag{4-66}$$

主、副控制器均采用比例积分调节规律，其计算式如前述。

仿真结果表明，$G_k(s)$ 的幅值过小时，即放大系数较小、积分时间较大时，系统跟踪特征较差，Y_m 与 Y 要较长时间才能无余差。若能在工作频率处满足 $1 + G_k(s)G_0(s)$ 与 $1 + G_k(s)G_m(s)$ 有相同的频率特性，则本方案的补偿效果也可较好，但鲁棒性要变差些。

仿真结果还表明，当 T_p、τ_p 大时，可选用大的 T_m 来补偿。由于采用 $\tau_m = 0$ 的方法，因此，采用比 T_p 大的 T_m 值来补偿动态特性的偏差。

与史密斯预估控制方案比较，可以看出，当 Y_m 与 Y 一致时，本方案的副控制器偏差为零，采用比例积分作用时，其输出可以为任意一个常数值。而史密斯预估控制方案的模型输入与系统对象的输入是同一个信号。由于本方案中，模型输入增加了一个可以为任意值的一个常量信号，它是根据系统输出与模型输出相比较而获得的校正信号，因此，本方案的控制作用可以增强，控制效果优于史密斯预估补偿控制方案。

4.3.2.3　定值控制系统

史密斯预估补偿控制系统对于随动控制能起到纯滞后的补偿作用。对于定值控制系统则因控制作用要比干扰的影响滞后 τ_p 的时间，因此，控制效果不如随动控制系统那样明显。

观测补偿控制方案用于定值系统时，采用图 4－26 所示框图：

系统的闭环特征方程是

$$1 + G_k(s)G_m(s) + G_c(s)G_0(s)G_k(s)G_m(s) = 0 \tag{4-67}$$

改写为：

$$G_c(s)G_0(s) = -1 - \frac{1}{G_k(s)G_m(s)} \tag{4-68}$$

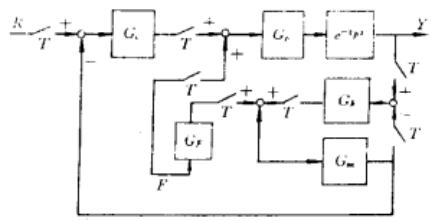


图 4－26　定值控制时的观测补偿控制系统

若采用单回路反馈控制，则对象为 $G_0(s)$ 时，闭环特征方程是：

$$G_c(s)G_0(s) = -1 \tag{4-69}$$

比较本方案和单回路控制方案，可以看到，系统的稳定条件发生变化。

单回路控制系统时，系统的稳定条件是沿着频率 ω 增加的方向，频率特性曲线 $G_c(j\omega)G_0(j\omega)$ 不包含判别点（－1，$j0$）。即（－1，$j0$）不在曲线的右边区域内。

本方案中，判别点由（－1，$j0$）成为判别点轨线 $-1-\frac{1}{G_k(j\omega)G_m(j\omega)}$。它是随 ω 变化而变化的轨线。若判别点轨线不被频率特性曲线包围则系统是稳定的。例如，若 $G_k(s)$、

$G_m(s)$ 均为纯比例环节，则因要组成员反馈，$K_kK_m > 0$，因此，判别点在$\left(-1-\frac{1}{K_kK_m}, j0\right)$处。它在$(-1, j0)$点的左侧。这表明，它的稳定性比单回路控制系统要好，当 $G_k(s)$ 的模较小时，$\left(-1-\frac{1}{K_kK_m}, j0\right)$较$(-1, j0)$更远，稳定性会更好。

根据不变性原理，干扰完全补偿的条件是

$$1 + G_k(s)G_m(s) - G_c(s)G_m(s)G_F = 0 \tag{4-70}$$

不失一般性，把 G_F 并入 $G_c(s)$ 中作为一项考虑，则有：

$$1 + G_k(s)G_m(s) - G_c(s)G_m(s) = 0$$

因此，满足干扰条件下定值控制要求的观测器应满足：

$$G_m(s) = \frac{1}{G_c(s) - G_k(s)} \tag{4-71}$$

上式表明，若主、副调节作用为纯比例，满足上式的观测器就能实现系统输出与设定的无余差定值控制要求。为了随动控制时也能无余差，通常，主、副调节作用选为比例、积分作用。即有

$$G_k(s) = K_k\left(1 + \frac{1}{T_ks}\right); G_c(s) = K_c\left(1 + \frac{1}{T_cs}\right);$$

由（4－71）式，求得 $G_m(s)$

$$G_m(s) = \frac{K_ms}{T_ms + 1} \tag{4-72}$$

式中，$T_m = \frac{K_c - K_k}{K_m}; K_m = \frac{T_cT_k}{K_cT_k - K_kT_c}$

若选择合适的控制器参数，使 $\frac{K_c}{T_c} = \frac{K_k}{T_k}$，则

$$G_m(s) = \frac{1}{K_c - K_k}$$

成为比例环节，实施十分方便。

采样控制系统仿真实施时，与上述相类似，但在观测器输入端，输入信号是副控制器输出与干扰经干扰通道输入的前馈信号相加。

4.3.3 解析预估补偿器

在史密斯预估补偿控制方案中，被控变量通过过程的模型来补偿过程的纯滞后。如果通过过程的模型来预估被控变量的未来值，并以这个预估值作为控制器的输入，这样也能有效地克服纯滞后的影响。这种控制方案就是解析预估补偿控制方案。它是 1969 年由莫尔（Moore）提出的。

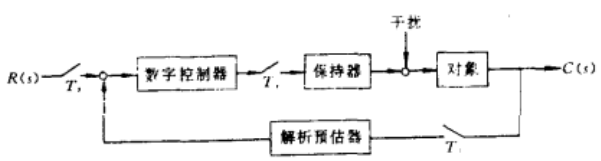


图 4－27　解析预估补偿控制系统原理图

4.3.3.1 解析预估补偿器的基本原理和结构

图 4－26 是解析预估补偿器控制系统的原理图。图中的 $e^{\tau_m s}$ 是通过求解预估方程来获得，而不是实际存在的超前环节。

由于是采样系统，因此，解析预估补偿器要预测的时间除了系统纯滞后 τ_p 外还应包括采样引入的半个采样周期时间。即

$$T' = \tau_p + \frac{1}{2}T_s = (N + \beta)T_s + 0.5T_s$$

或

$$\frac{T'}{T_s} = N + \beta + 0.5$$

因此，若解析预估补偿器能通过解析的方法求出当前时刻 t 以后 $T' + t$ 时刻的输出 $y(t + T')$，那么，控制器的测量值与史密斯预估补偿控制时的 B 点测量值相同。这时，闭环特征方程将不含有纯滞后项。

4.3.3.2 解析预估补偿控制方案的实施

假设对象 $G_p(s) = \dfrac{K_p}{T_p s + 1}$，纯滞后为 τ_p。而采用的预估模型为 $G_m(s) = \dfrac{K_m}{T_m s + 1}$，其纯滞后为 τ_m。根据控制器的输出 u_k，求出预估模型的输出 $\hat{y}_{k+T'}$ 是

$$\hat{y}_{k+T'} = A_1\{A_3A_2^N y_k + A_2^N K_m(1 - A_3)u_{k-(N+1)} + K_m(1 - A_2)\sum_{i=1}^{N}A_2^{i-1}u_{k-i}\} + K_m(1 - A_1)u_k \tag{4-73}$$

式中，$A_1 = e^{-\frac{T_s}{2T_m}}$。$A_2$ 与 A_3 的定义见（4－50）式。式中的预估模型输入 u_k 由下式求得：

$$u_k = K_c(R_k - \hat{y}_{k+T'}) \tag{4-74}$$

这是采用比例作用时的情况。可以看到，系统的余差是

$$\lim_{t\to\infty}|R(t) - y(t)| = \frac{1}{1 + K_cK_m}$$

为此，可用设定值的修正来消除，即令

$$\rho_k = \frac{K_cK_m + 1}{K_cK_m}R_k \tag{4-75}$$

则预估模型输入 u_k 成为：

$$u_k = K_c(\rho_k - \hat{y}_{k+T'}) \tag{4-76}$$

由于实际系统的参数和模型参数的不一致，因此，预估值 $\hat{y}$ 与实测值 y 之间会有偏差，其结果是系统实际输出与设定值存在偏差。一种简单的解决办法是把 y_k 与 $\hat{y}_k$ 的偏差，作为一个扰动量，对闭环系统进行修正。

设扰动的估计量 $\hat{d}_k$ 由下式递推。

$$\hat{d}_k = \hat{d}_{k-1} + M(y_k - \hat{y}_k) \tag{4-77}$$

扰动量初值 $\hat{d}_0 = 0$。

解析预估值 $\hat{y}_k$ 由下式给出：

$$\hat{y}_k = A_2y_{k-1} + K_m\left(1 - \frac{A_2}{A_3}\right)[u_{k-(N+1)} + \hat{d}_{k-1}] + K_m\left(\frac{A_2}{A_3}\right)(1 - A_3)[u_{k-(N+2)} + \hat{d}_{k-1}] \tag{4-78}$$

而 $\hat{y}_{k+T'}$ 由下式确定：

$$\hat{y}_{k+T'} = A_1A_3A_2^N y_k + A_1A_2^N K_m(1 - A_3)[u_{k-(N+1)} + \hat{d}_k]$$

$$+ A_1K_m(1 - A_2)\sum_{i=1}^{N}A_2^{i-1}(U_{k-i} + \hat{d}_k)$$

$$+ K_m(1 - A_1)(u_k + \hat{d}_k) \tag{4-79}$$

控制器输出 u_k 是：

$$u_k = C_1R_k - C_2y_k - C_3[u_{k-(N+1)} + \hat{d}_k]$$

$$- C_4\sum_{i=1}^{N}A_2^{i-1}(u_{k-i} + \hat{d}_k) - C_5\hat{d}_k - \hat{d}_k \tag{4-80}$$

式中，$C_1 = \dfrac{K_cK_m + 1}{K_m[1 + K_cK_m(1 - A_1)]}$

$$C_2 = \frac{K_cA_1A_2^NA_3}{1 + K_cK_m(1 - A_1)}$$

$$C_3 = \frac{K_cK_mA_1A_2^N(1 - A_3)}{1 + K_cK_m(1 - A_1)}$$

$$C_4 = \frac{K_cK_mA_1(1 - A_2)}{1 + K_cK_m(1 - A_1)}$$

$$C_5 = \frac{K_cK_m(1 - A_1)}{1 + K_cK_m(1 - A_1)}$$

由于在 y_k 与 $\hat{y}_k$ 不一致时，被估计的干扰会随时间连续变化，构成了积分作用，从而消除了余差。

4.4 采 样 周 期

4.4.1 不同采样周期对模型参数的影响

在采样控制系统中，有两种场合需要采用不同的采样周期。

①控制系统本身由不同的采样周期对不同的测量值进行采样。例如在温度和流量组成的串级控制系统中，流量的采样周期是 1 秒，温度的采样周期是 5 秒，这样的采样系统就有两个不同的采样周期。

②对同一个被仿真的系统，为了提高精度或者加快运算速度，需要更改仿真所采用的采样周期。

对于第一种情况，常采用小采样周期对对象仿真若干步后再用大采样周期仿真一步的内外嵌套的方法。对采用大采样周期的对象，若需要增加时间点上的采样值，则可与第二种情况相同的方法处理。

对于第二种情况，可采用与根匹配法相类似的方法处理，现叙述如下。

设在采样周期 T_s 时获得的模型 $G(z)$ 是

$$G(z) = \frac{K_z(z - z_1)(z - z_2)\cdots(z - z_m)}{(z - p_1)(z - p_2)\cdots(z - p_n)} \tag{4-81}$$

现改用采样周期 T'_s，相应模型更改为

$$G'(z) = \frac{K'_z(z - z'_1)(z - z'_2)\cdots(z - z'_m)}{(z - p'_1)(z - p'_2)\cdots(z - p'_n)} \tag{4-82}$$

由于不同采样周期时的两个模型具有相同的特性，即映射到 s 平面上应具有相同的零极点。设零、极点为 s_{z_1}、…、s_{z_m} 和 s_{p_1}、…、s_{p_n}。根据映射的关系有：

$$z_i = e^{T_s z_i};\ p_j = e^{T_s p_j} \qquad (i = 1,\cdots;m;j = 1,\cdots,n)$$
$$z'_i = e^{T'_s z_i};\ p'_j = e^{T'_s p_j} \qquad (i = 1,\cdots,m;j - 1,\cdots,n)$$

联立求解得

$$z'_i = e^{\frac{T'_s}{T_s}\ln z_i};\ p'_j = e^{\frac{T'_s}{T_s}\ln p_j} \quad (i = 1,\cdots,m;j = 1,\cdots,n) \tag{4-83}$$

而 K'_z 的确定可与根匹配法相同，用具有相同的稳态值或频率特性来确定。

一种简易的零极点更换的方法是用双线性变换公式进行映射。即：

$$S = \frac{2}{T_s}\frac{z-1}{z+1}$$

相应的零、极点计算公式如下：

$$z'_i = \frac{T_s(z_i+1) + T'_s(z_i-1)}{T_s(z_i+1) - T'_s(z_i-1)} \qquad (i = 1,\cdots,m)$$
$$p'_j = \frac{T_s(p_j+1) + T'_s(p_j-1)}{T_s(p_j+1) - T'_s(p_j-1)} \qquad (j = 1,\cdots,n) \tag{4-84}$$

〔例 4-12〕 在采样周期 $T_s = 0.04$ 时仿真模型 $G(z)$ 是 $G(z) = 2.62\frac{z-0.98}{z-0.64}$，试求改变采样周期成为 $T'_s = 0.1$ 时的 $G'(z)$。

根据（4-83）式，求得：

$$z'_1 = e^{\frac{0.1}{0.04}\ln 0.98} = 0.9508;\ p'_1 = e^{\frac{0.1}{0.04}\ln 0.64} = 0.3277$$

因此，

$$G'(z) = K'_z\frac{z-0.9508}{z-0.3277}$$

输入阶跃信号时，有相同的稳态值可得：

$$\lim_{t\to\infty} y(t) = \lim_{z\to 1}\left\{\frac{z-1}{z}\cdot G(z)\cdot\frac{z}{z-1}\right\} = 2.62\cdot\frac{1-0.98}{1-0.64} = 0.14555$$

同样地，有

$$\lim_{z\to 1}\left\{\frac{z-1}{z}G'(z)\frac{z}{z-1}\right\} = K'_z\frac{1-0.9508}{1-0.3277} = 0.14555$$

因此，有

$$K'_z = 1.9889;\ G'(z) = 1.9889\frac{z-0.9508}{z-0.3277}$$

若采用双线性变换公式，有

$$z'_1 = \frac{T_s(z_1+1) + T'_s(z_1-1)}{T_s(z_1+1) - T'_s(z_1-1)}$$
$$= \frac{0.04(0.98+1) + 0.1(0.98-1)}{0.04(0.98+1) - 0.1(0.98-1)} = 0.9507$$
$$p'_1 = \frac{0.04(0.64+1) + 0.1(0.64-1)}{0.04(0.64+1) - 0.1(0.64-1)} = 0.2913$$

因此，

$$G'(z) = K'_z\frac{z-0.9507}{z-0.2913}$$
$$\lim_{z\to 1}\left\{\frac{z-1}{z}G'(z)\cdot\frac{z}{z-1}\right\} = K'_z\frac{1-0.9507}{1-0.2913} = 0.14555$$

$$K'_z = 2.092; G'(z) = 2.092\frac{z - 0.9507}{z - 0.2913}$$

4.4.2 不同采样周期时系统的稳定性分析

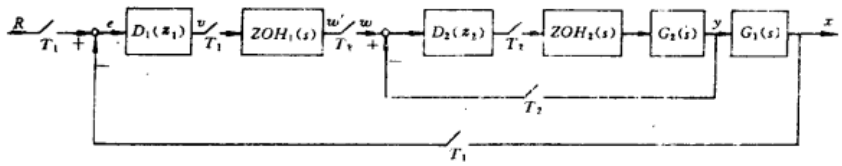


图 4-28 串级采样控制系统

下面以单变量多回路系统为例说明当回路中的采样周期取不同数值时系统的稳定性。图 4-27 是典型的串级采样控制系统。其中内环采用较小的采样周期 T_2，外环采用较大的采样周期 T_1，通常二者有整数倍的关系，即

$$T_1 = lT_2$$

由于采用不同的采样周期，因此，为区分相应的 z 变换，分别用 z_1、z_2 表示相应的算子。因此，外环的主控制器脉冲传递函数用 $D_1(z_1)$ 表示。内环的副控制器脉冲传递函数用 $D_2(z_2)$ 表示。

若系统的内环用 $\frac{X(z_2)}{W(z_2)}$ 表示。并用差分方程描述为：

$$\begin{aligned} x(kT_2) = &- a_1x[(k-1)T_2] - a_2x[(k-2)T_2] - \cdots - a_mx[(k-m)T_2] \\ &+ b_1w[(k-1)T_2] + b_2w[(k-2)T_2] + \cdots + b_mw[(k-m)T_2] \end{aligned} \quad (4-85)$$

下面分几种不同情况进行讨论。

(1) 外环没有数字校正环节

这时，$D_1(z_1) = 1$，表示主控制器是 1:1 比例作用。分下列三种情况讨论稳定性问题。

①$l > m$。由于内、外环采样开关同步，并有 $T_1 = lT_2$，因此，把内、外环时间序列放在同一坐标系中进行分析，它们通过零阶保持器 1 发生联系。即

$$v(nT_1) = e(nT_1) \qquad (\because D_1(z_1) = 1) \quad (4-86)$$

$$v(nT_1) = v(nlT_2) \quad (4-87)$$

在 lnT_2 到 $lnT_2 + (l-1)T_2$ 的 l 个采样时刻零阶保持器的输出 $w(lnT_2 + iT_2)$ 保持不变，($i = 0, \cdots, l-1$)。其值等于 $v(nT_1)$。即

$$w(nlT_2 + iT_2) = v(nT_1) = v(nlT_2) = e(nlT_2) \qquad i = 0, \cdots, l-1 \quad (4-88)$$

考虑到零阶保持器 1 的周期性保持的特点，对 (4-85) 式中的 k 可分别取下列 l 个数值，得到 l 个差分方程。

$$k = ln, ln+1, \cdots, ln+l-1 \quad (4-89)$$

为此，把 (4-85) 改写成：

$$\begin{aligned} x[(k+m)T_2] = &- a_1x[(k+m-1)T_2] - \cdots - a_mx[(kT_2)] \\ &+ b_1w[(k+m-1)T_2] + \cdots + b_mw[kT_2] \end{aligned} \quad (4-90)$$

主环的偏差：$e(nT_1) = u(nT_1) - x(nT_1)$，或写成：

$$e(nlT_2) = u(nlT_2) - x(nlT_2) \quad (4-91)$$

由于 $l > m$，因此，把 l 个 k 的取值 ((4-89) 式给出) 代入 (4-90) 式，并把 (4-88) 与 (4-91) 式也代入，可以消去 e 与 w 项，得到规一化的系统状态方程，它以采样周

期 T_2 进行采样。为简便，下面省略 T_2，不记入式内。

$$\begin{aligned}
x(ln+m) &= -a_1x(ln+m-1)-\cdots-am(b_1+\cdots+b_m)x(ln)\\
&\quad+(b_1+\cdots+b_m)u(ln)\\
x(ln+m+1) &= -a_1x(ln+m)-\cdots-a_mx(ln+1)\\
&\quad-(b_1+\cdots+b_m)x(ln)+(b_1+\cdots+b_m)u(ln)\\
&\cdots\cdots\\
x(ln+l-1) &= -a_1x(ln+l-2)-\cdots-a_mx(ln+l-m-1)\\
&\quad-(b_1+\cdots+b_m)x(ln)+(b_1+\cdots+b_m)u(ln)\\
x(ln+l) &= -a_1x(ln+l-1)-\cdots-a_mx(ln+l-m)\\
&\quad-(b_1+\cdots+b_m)x(ln)+(b_1+\cdots+b_m)u(ln)\\
&\cdots\cdots\\
x(ln+l+m-1) &= -a_1x(ln+l+m-2)-\cdots\\
&\quad-a_mx(ln+l-1)+(b_l+b_2+\cdots+b_m)u(ln+l)+b_mu(ln)\\
&\quad-b_mx(ln)
\end{aligned} \tag{4-92}$$

写成矩阵形式，即为：

$$\begin{cases} A\ddot{x}_n+B\dot{x}_n=Cu_n \\ D\dot{x}_{n+1}+E\ddot{x}_n+F\dot{x}_n=Gu_n \end{cases} \tag{4-93}$$

式中，A、B、C、D、E、F、G 均为（4－92）中相应项系数。

$$\ddot{x}_n=\begin{bmatrix} x(ln+m) \\ \vdots \\ x(ln+l-1) \end{bmatrix};\dot{x}_n=\begin{bmatrix} x(ln) \\ \vdots \\ x(ln+m-1) \end{bmatrix};x_n=\begin{bmatrix} \dot{x}_n \\ \ddot{x}_n \end{bmatrix}$$

$$u_n=\begin{bmatrix} u(ln) \\ \vdots \\ u(ln+l) \end{bmatrix};\dot{x}_{n+1}=\begin{bmatrix} x(ln+l) \\ \vdots \\ x(ln+l+m-1) \end{bmatrix}; \tag{4-94}$$

由于 A、D 矩阵是对角线元素为 1 的下三角矩阵，故可求逆。（4－93）因此可简化为：

$$\begin{cases} \ddot{x}_n+A^{-1}B\dot{x}_n=A^{-1}Cu_n \\ \dot{x}_{n+1}+D^{-1}(F-EA^{-1}B)\dot{x}_n=D^{-1}(G-EA^{-1}C)u_n \end{cases} \tag{4-95}$$

当 $u_n=0$ 得系统状态方程的齐次方程是

$$\begin{cases} \ddot{x}_n+A^{-1}B\dot{x}_n=0 \\ \dot{x}_{n+1}+D^{-1}(F-EA^{-1}B)\dot{x}_n=0 \end{cases} \tag{4-96}$$

根据（4－96）式，可知，系统稳定的充要条件是：系数矩阵 $D^{-1}(F-EA^{-1}B)$ 的所有特征值的模都小于 1。

②$l=m$。它是 $l>m$ 的特例。这时（4－95）式退化为一个方程：

$$D\dot{x}_{n+1}+F\dot{x}_n=Gu_n \tag{4-97}$$

与上述情况相类似，得到系统稳定的充要条件是 $D^{-1}F$ 矩阵所有特征值的模都小于 1。

③$l<m$。可以用上述方法推理，得到类似的稳定判据。

(2) 外环有数字校正

可假设 $D_1(z_1)=\dfrac{d_0+d_1z_1^{-1}}{1+c_1z_1^{-1}}$

这种情况更接近实际系统，但分析更为复杂，以 $l>m$ 的情况说明。假如取 $l=4$，$m=2$。这时系统可用下列差分方程描述：

$$\begin{aligned}
&x(k+2)=-a_1x(k+1)-a_2x(k)+b_1w(k+1)+b_2w(k)\\
&w(4n)=w(4n+1)=w(4n+2)=w(4n+3)=v(4n)\\
&v(4n+4)=-c_1v(4n)+d_0e(4n+4)+d_1e(4n)\\
&e(4n)=u(4n)-x(4n)
\end{aligned}\tag{4-98}$$

以 $k=4n$，$4n+1$，$4n+2$，$4n+3$ 代入 (4-98)，并消去中间变量 v，w 及 e，得到下列方程组。

$$\begin{aligned}
x(4n+6)=&-[a_1x(4n+5)+(a_2+b_1d_0+b_2d_0)x(4n+4)+c_1x(4n+2)\\
&+c_1a_1x(4n+1)+(a_2+b_1d_1+b_2d_1)x(4n)]\\
&+(b_1+b_2)d_0u(4n+4)+(b_1+b_2)d_1u(4n)
\end{aligned}\tag{4-99}$$

$$\begin{aligned}
x(4n+7)=&-[a_1x(4n+6)+a_2x(4n+5)+(b_1d_0+b_2d_0)x(4n+4)+c_1x(4n+3)\\
&+c_1a_1x(4n+2)+c_1a_2x(4n+1)+(b_1d_1+b_2d_1)x(4n)]\\
&+(b_1+b_2)d_0u(4n+4)+(b_1+b_2)d_1u(4n)
\end{aligned}\tag{4-100}$$

$$\begin{aligned}
x(4n+8)=&-[a_1x(4n+7)+a_2x(4n+6)+(c_1+b_1d_0+b_2d_0)x(4n+4)\\
&+c_1a_1x(4n+3)+c_1a_2x(4n+2)+(b_1d_1+b_2d_1)x(4n)]\\
&+(b_1+b_2)d_0u(4n+4)+(b_1+b_2)d_1u(4n)
\end{aligned}\tag{4-101}$$

$$\begin{aligned}
x(4n+9)=&-[(a_1+b_1d_0)x(4n+8)+a_2x(4n+7)+c_1x(4n+5)\\
&+(c_1a_1+b_1d_1+b_2d_0)x(4n+4)]+c_1a_2x(4n+3)+b_2d_1x(4n)]\\
&+b_1d_0u(4n+8)+(b_1d_1+b_2d_0)u(4n+4)+b_2d_1u(4n)
\end{aligned}\tag{4-102}$$

可以改写成：

$$\begin{aligned}
&A\begin{bmatrix}x(4n+6)\\x(4n+7)\\x(4n+8)\\x(4n+9)\end{bmatrix}+B\begin{bmatrix}x(4n+2)\\x(4n+3)\\x(4n+4)\\x(4n+5)\end{bmatrix}+C\begin{bmatrix}x(4n-2)\\x(4n-1)\\x(4n)\\x(4n+1)\end{bmatrix}\\
&=Du(4n+8)+Eu(4n+4)+Fu(4n)
\end{aligned}\tag{4-103}$$

系数矩阵 A、B、C、D、E、F 均可由差分方程求得。由于 A 矩阵可求逆，(4-103) 式可改写成：

$$\begin{bmatrix} x(4n+2) \\ \vdots \\ x(4n+5) \\ x(4n+6) \\ \vdots \\ x(4n+9) \end{bmatrix} + \left[\begin{array}{c|c} 0 & -I \\ \hline A^{-1}C & A^{-1}B \end{array}\right] \begin{bmatrix} x(4n-2) \\ \vdots \\ x(4n+1) \\ x(4n+2) \\ \vdots \\ x(4n+5) \end{bmatrix} = \left[\begin{array}{c} 0 \\ \hline A^{-1}D \end{array}\right] u(4n+8)$$

$$+ \left[\begin{array}{c} 0 \\ \hline A^{-1}E \end{array}\right] u(4n+4) + \left[\begin{array}{c} 0 \\ \hline A^{-1}F \end{array}\right] u(4n) \tag{4-104}$$

因此，令 $u=0$ 可得系统稳定的充要条件是矩阵 H

$$H = \left[\begin{array}{c|c} 0 & -I \\ \hline A^{-1}C & A^{-1}B \end{array}\right] \tag{4-105}$$

的所有特征值的模都小于 1。

对于 $l=m$ 及 $l<m$ 的情况，可参照本方法进行分析。

4.4.3 采样周期的选择

采样定理确定了采样频率的下限值。为了复现原信号，采样周期有一个上限值。即：

$$\omega_s < 2\omega_m \text{ 或 } T_s > \frac{T_m}{2}$$

式中 ω_m、T_m 是系统的最高工作频率和周期。

采样周期的下限值与所选用的计算机系统有关。例如，在积分运算时，由于采样周期 T_s 太小，以致两次采样值所得结果造成积分输出的增量为机器零。这就形成积分不灵敏区。通常，在二进制小数位选择正确、计算机字长是 16 位时，对于 *PI* 及 *PID* 控制，采样周期的下限值是：

$$T_s > \frac{T_i}{100} \tag{4-106}$$

式中，T_i 是积分时间。

如果已知过程模型的动态参数，则可以得到所需采样周期的较好的估计值。通常选用符合控制性能要求下较大的采样周期。

应该指出，在采样控制系统中，由于引入采样开关和保持器，使系统引入了附加的滞后，其数值是采样周期的一半。当采样周期大时，引入的附加滞后也大。因此采样控制系统的控制性能不如相应的连续控制系统。

选择采样周期最简单的方法是根据 1963 年用户会议制定的 DDC 指南中所推荐的方法。推荐的方法是：对于流量控制回路，采样周期为 1s；液位和压力控制回路为 5s；温度和成分控制回路为 20s。

对于随动控制系统，采样周期的选择很大程度上依赖于系统的动态性能。一般情况下，系统的闭环频率特性具有低通滤波的特性。当随动系统的输入信号频率高于谐振频率 ω_0 时，将会很快地衰减。而且 $\omega_0 \approx \omega_c$，$\omega_c$ 是开环频率特性的截止频率。因此，通过对系统频率特性的分析，可以看到，高于 ω_c 的频率分量会被很快衰减掉。所以，常在随动控制系统中选用采样频率 $\omega_s \approx 10\omega_c$，这时由保持器引入的相位移很小，不会产生采样信号的频谱混迭，同时能保证有较满意的动态性能。

4.5 采样控制系统数字仿真程序

4.5.1 一阶滞后环节的数字仿真程序

当对象用一阶滞后环节描述时，传递函数表示为：

$$G(s) = \frac{K_p}{T_p s + 1} e^{-\tau_p s}$$

采样周期 T_s，采用零阶保持器及改进 z 变换法，得到 $HG(z)$

$$HG(z) = \frac{K_p(c_1 + c_2 z^{-1}) z^{-(d+1)}}{(1 - bz^{-1})}$$

式中的参数见(4－31)式。对应的数字仿真程序如下所示。

```
10  REM SAMPLING SYSTEM(1 ORDER)
20  DIM YY(600),O(200),E(200),A(50),B(50)
30  CLS
40  LOCATE 2,20
50  PRINT"Reading Data!"
60  REM PR,Kp,T,DT,ST,H,N,A(I),B(I)
70  READ PR,KP,T,DT,ST,H,N
80  FOR I=1 TO N
90  READ A(I):REM Denominator of D(z)
100  NEXT I
110  FOR I=0 TO N
120  READ B(I):RDM Numerator of D(z)
130  NEXT I
200  SP=1
210  M=DT/ST
220  SH=ST/H
230  FOR I=1 TO DT/H
240  YY(I)=0
250  NEXT I
260  FOR I=1 TO N
270  O(I)=0
280  NRXT I
290  MM=1-(DT-INT(DT/H)*H)/H
300  BB=EXP(-H/T)
310  C1=1-EXP(-MM*H/T)
320  C2=EXP(-MM*H/T)-BB
330  SCREEN 2,0
340  CLS
350  KEY OFF
```

```
360  LINE(20,20)-(20,170)
370  LINE -(620,170)
380  FOR I=SH TO 600 STEP SH
390  FOR J=1 TO 15
400  LINE(I+20,J*10+18)-(20+I,J*10+20)
410  NEXT J
420  NEXT I
500  E(0)=SP-YY(DT/H)
510  R=0
520  Z=0
530  FOR I=1 TO N
540  R=R-A(I)*O(I)
550  NEXT I
560  FOR I=0 TO N
570  R=R+B(I)*E(I)
580  NEXT I
590  O(0)=R
600  IF O(0)>MAX THEN MAX=O(0)
610  LINE(20+G,90-70*O(0)/MAX)-(20+G,90-70*O(1)/MAX)
620  FOR I=N-1 TO 0 STEP -1
630  E(I+1)=E(I)
640  O(I+1)=O(I)
650  NEXT I
660  LOCATE 1,1
670  PRINT"
680  LOCATE 1,1
690  IF PR=1 THEN LPRINT G*H;TAB(20);YY(DT/H);TAB(40);0(O)
700  YY(0)=BB*YY(1)+C1*KP*O(M-1)+C2*KP*O(M-1)
710  G=G+1
720  PSET (20+G,160-70*YY(DT/H))
730  PSET (20+G,90-70*O(0)/MAX)
740  FOR J=DT/H-1 TO 0 STEP -1
750  YY(J+1)=YY(J)
760  NEXT J
770  IF G>600 THEN 810
780  Z=Z+1
790  IF Z<SH THEN 660
800  GOTO 500
810  LOCATE 1,1
820  PRINT"
```

```
830  LPRINT CHR$(27)"W"
840  END
```

程序运行前需按下列次序依次送入有关参数：

$$PR、K_p、T、DT、ST、H、A(I)、B(I)$$

其中，各参数定义如下：

PR：打印数据标志，PR＝1 则打印数据。

K_p：对象放大系数。

T：对象时间常数，即 T_p。

DT：对象纯滞后时间，即 τ_p。

ST：采样系统采样时间，即 T_s。

H：步长。

N：数字控制器 $D(z)$的阶次。

A(I)：$D(z)$的分母多项式系数，$I=1,2,\cdots,N$。

B(I)：$D(z)$的分子多项式系数，$I=0,1,\cdots,N$。

$$D(z)=\frac{B(z)}{A(z)}=\frac{B(0)+B(1)z^{-1}+\cdots+B(N)z^{-N}}{1+A(1)z^{-1}+A(2)z^{-2}+\cdots+A(N)z^{-N}}$$

根据 z 域设计法确定D（z）的分子、分母多项式系数。例如对于大林算法，有

$$D(z)=\frac{(1-\gamma)(1-bz^{-1})}{K_p(c_1+c_2z^{-1})[1-\gamma z^{-1}-(1-\gamma)z^{-(d+1)}]}$$

〔例 4－13〕已知对象 $G(s)=\dfrac{e^{-1.46s}}{3.3s+1}$，采样周期 $T_s=1$，求大林算法，改进大林算法及卡尔曼算法的闭环输出。

按表 4－2 求得大林算法的数字控制器 D（z）是：

$$\begin{aligned}D(z)&=\frac{2.607(1-0.7386z^{-1})}{(1+0.7323z^{-1})(1-z^{-1})(1+0.3935z^{-1})}\\&=\frac{2.607-1.9255z^{-1}}{1+0.1258z^{-1}-0.8376z^{-2}-0.2882z^{-3}}\end{aligned}$$

因此，输入下列语句：

```
100  DATA 0,1,3.3,1.46,1,0.05
1010 DATA 3,0.1258,-0.8376,-0.2882,2.607,-1.9255,0,0
```

程序运行后显示图 4－28 所示的闭环响应曲线。上部曲线是数字控制器输出，下部曲线是对象输出。垂直的虚线表示各个采样时刻。可以看到，数字控制器输出有明显跳动现象。为此，对 $z=-0.7323$ 的极点用 $z=1$ 代替，得改进大林算法的数字控制器 D（z）：

$$D(z)=\frac{1.5094-1.11155z^{-1}}{1-0.6065z^{-1}-0.3935z^{-2}}$$

为此，只需对原有程序中的 1010 语句更改为：

```
1010 DATA 2,-0.6065,-0.3935,1.5049,-1.11155,0
```

运行显示于图 4－29，可见，数字控制器输出跳动明显改善，对象输出也比较平稳。

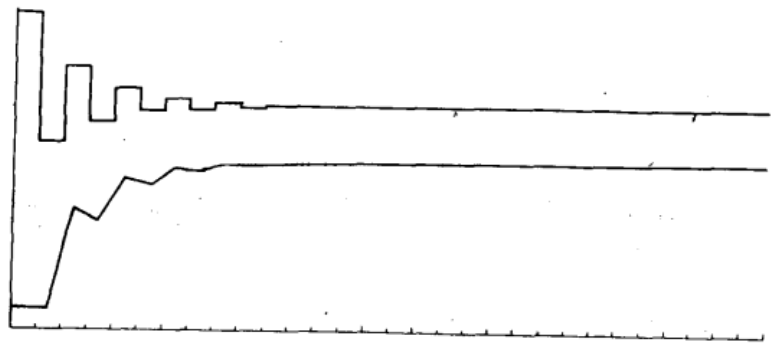


图 4－29　大林算法闭环输出

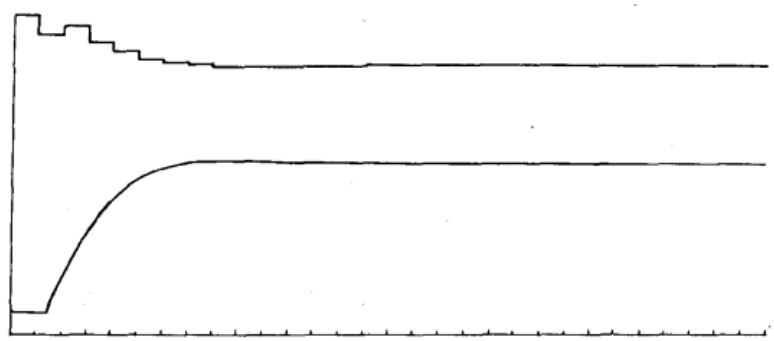


图 4－30　改进大林算法闭环输出

按表 4－3 的公式，求出卡尔曼算法的数字控制器 $D(z)$ 是：

$$D(z) = \frac{3.826 - 2.8259z^{-1}}{1 - 0.5773z^{-2} - 0.4242z^{-3}}$$

改变 1010 语句为：

1010　DATA 3,0, －0.5773, －0.4242,3.826, －2.8259,0,0

运行结果示于图 4－30。

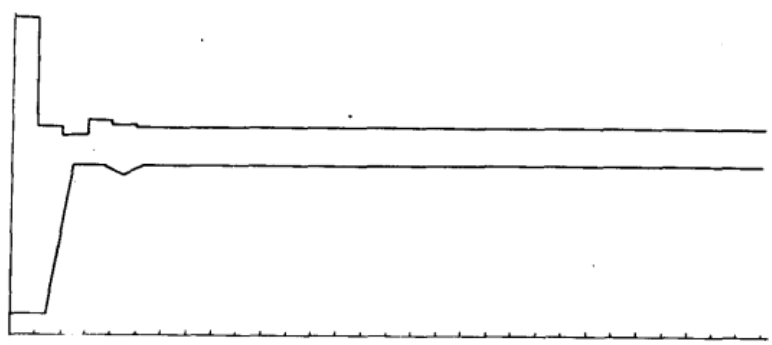


图 4－31　卡尔曼算法闭环输出

4.5.2 二阶滞后环节的数字仿真程序

当对象是二阶滞后环节描述时，传递函数表示为：

$$G(s) = \frac{K_p}{(T_1 s + 1)(T_2 s + 1)} e^{-\tau_p s}$$

采样周期 T_s，同样有$HG(z)$

$$HG(z) = \frac{K_p(c_3 + c_4 z^{-1}) z^{-(d+1)}}{(1 - \alpha z^{-1})(1 - \beta z^{-1})}$$

数字仿真程序与一阶滞后环节数字仿真程序相类似，只需对 $HG(z)$部分作相应更改。数字控制器 $D(z)$则根据 z 域设计法确定。

〔例 4－14〕已知对象 $G(s) = \frac{e^{-s}}{(3s+1)(5s+1)}$，采样周期 $T_s = 1$，求 V.E 算法时的闭环输出。

根据例 4－9 的计算结果，有：

$$D(z) = \frac{12.3 - 18.884 z^{-1} + 7.2152 z^{-2}}{1 - 0.368 z^{-1} - 0.344 z^{-2} - 0.288 z^{-3}}$$

运行结果示于图 4－31。

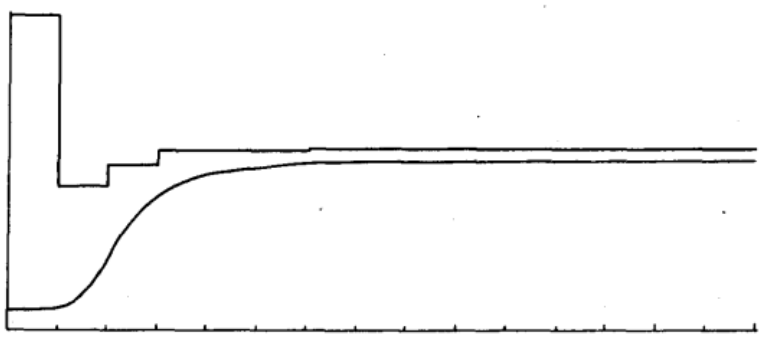


图 4－32　V.E 控制算法闭环输出

数字仿真程序中，控制器部分是按 $D(z)$计算的。而对象部分是按 $HG(z)$计算的，即未采用数值积分法求解，为了使两部分联系起来，采用了对象做若干步(程序中 ST/H 表示)，再做一步控制器输出计算的方法。

4.5.3 史密斯预估补偿控制数字仿真程序

假设对象为

$$G_p(s) e^{-\tau_p s} = \frac{K_p}{T_p s + 1} e^{-\tau_p s}$$

模型为

$$G_m(s) = \frac{K_m}{T_m s + 1} e^{-\tau_m s}$$

采样周期 T_s。则史密斯预估补偿控制方案采用图 4－21 方式实施。根据例 4－10 可编制相应的数字仿真程序如下：

```
10  REM SMITH PREDICTOR
50  REM KP, TP, TOP, KM, TM, TOM
```

```
60  READ KP, TP, TOP, KM, TM, TOM
70  REM TS, KC, TI, TD
80  READ TS, KC, TI, TD
100  CLS
110  SCREEN 2, 0
120  KEY OFF
130  LINE(20, 10) - (20, 160)
140  LINE - (620, 160)
150  FOR I = 1 TO 600/TOP * TS
160  PSET(20 + I * TOP/TS, 159)
165  PSET(20 + I * TOP/TS, 158
170  NEXT I
180  FOR I = 0 TO 9
190  PSET(21, 10 + 15 * I)
200  FOR J = 1 TO 600/TOP * TS
210  PSET(20 + J * TOP/TS, 10 + 15 * I)
220  NEXT J
230  NEXT I
240  LOCATE 2, 1
250  PRINT"2"
260  LOCATE 21, 1
270  PRINT"0"
280  LOCATE 22, 77
290  PRINT 600/TOP * TS
300  LOCATE 22, 3
310  PRINT"0"
320  LOCATE 11, 1
330  PRINT"1"
340  LOCATE 22, 39
350  PRINT 300/TOP * TS
480  KI = KC/TI * TS
490  KD = KC * TD/TS
500  K1 = KC * 2 + KI * 2 + KD
510  K2 = - KC - KI - KD
520  K3 = KI
1000  REM MAIN PROGRAM
1010  A1 = INT(TOP/TS)
1020  A6 = INT(TOM/TS)
1030  B1 = TOP/TS - A1
1040  B6 = TOM/TS - A6
```

```
1050  A4 = EXP( - TS/TP)
1060  A2 = EXP( - TS/TM)
1070  A5 = EXP( - B1 * TS/TP)
1080  A3 = EXP( - B6 * TS/TM)
1090  M = A1
1100  IF A6 > A1 THEN M = A6
1110  DIM Y(M = 2), YY(M + 2), U(M + 2), E(2), B(1)
1120  SP = 1 : G = 0
1130  Y(0) = A4 * Y(1) + A4 * KP * (1/A5 - 1) * U(A1 + 2) + KP * (1 - A4/A5) * U
      (A1 + 1)
1140  YY(0) = A2 * YY(1) + A2 * KM * (1/A3 - 1) * U(A6 + 2) + KM * (1 - A2/A3)
      * U(A6 + 1)
1150  B(0) = KM * (1 - A2) * U(1) + A2 * B(1)
1160  E(0) = SP - Y(0) - B(0) + YY(0)
1170  FOR I = M + 1 TO 0 STEP  - 1
1180  Y(I + 1) = Y(I)
1190  YY(I + 1) = YY(I)
1200  U(I + 1) = U(I)
1210  NEXT I
1220  B(1) = B(0)
1230  EE(0) = EE(0) + E(0)
1240  E1(0) = E(2) - 2 * E(1) + E(0)
1250  U(0) = KC * E(0) + KI * EE(0) + KD * E1(0)
1260  E(2) = E(1) : E(1) = E(0)
1270  LOCATE 1, 10
1280  PRINT G * TS/TOP, Y(0)
1290  PSET(20 + G, 160 - 75 * Y(0))
1300  REM PSET(20 + G, 160 - 75 * YY(0))
1310  G = G + 1
1320  IF G < 600 THEN 1130
1330  LOCATE 1, 10
1340  PRINT"
1350  LOCATE 23, 1
1360  PRINT"Kc = "; KC
1370  LOCATE 23, 15
1380  PRINT"Ti = "; TI
1390  LOCATE 23, 30
1400  PRINT "Td = "; TD
1410  LOCATE 1, 15
1420  PRINT "Km = "; KM
```

```
1430  LOCATE 1,30
1440  PRINT "Tm=";TM
1450  LOCATE 1,45
1460  PRINT "Tor m=";TOM
1470  LPRINT CHR$(27)"W"
1480  END
```

程序需依下列次序用 DATA 语句输入有关参数：K_p、T_p、τ_p、K_m、T_m、τ_m、T_s、K_c、T_i、T_d。

K_p：对象放大系数。

K_m：模型放大系数。

T_p：对象时间常数。

T_m：模型时间常数。

τ_p：(TOP)：对象纯滞后。

τ_m：(TOM)：模型纯滞后。

T_s：采样周期。

K_c：调节器增益。

T_i：调节器积分时间。

T_d：调节器微分时间。

K_i、K_d 用于 *PID* 控制时的积分项增益和微分项增益。即有：

$$K_i = \frac{K_c}{T_i} \cdot T_s \quad ; \quad K_d = \frac{K_c T_d}{T_s}$$

K_1、K_2、K_3 用于预测 *PID* 控制时的数字仿真。按（4－13）式，有：

$$K_1 = 2K_c + 2K_i + K_d; K_2 = -K_c - K_i - K_d; K_3 = K_i$$

用于预测 *PID* 控制时，程序中，1250 语句改为：

```
1250  U(0)=K1*E(0)+K2*E(1)+K3*EE(0)
```

本程序也可用于

$$G_p(s)e^{-\tau_p s} = \frac{K_p}{T_p s + 1} e^{-\tau_p s}$$

采用 *PID* 或预测 *PID* 控制的场合，以便与史密斯预估补偿控制方案进行比较，为此，只要在 1160 语句改为：

```
1160  E(0)=SP-Y(0)
```

〔例 4－15〕　当 $K_p=0.5$，$T_p=5$，$\tau_p=1$ 时，比较下列情况的闭环输出。

①完全补偿时，即 $K_m=0.5$，$T_m=5$，$\tau_m=1$。

②不完全补偿时，采用 *PID* 控制和预测 *PID* 控制算法的结果。

当完全补偿时，送入数据语句如下：

```
5000  DATA 0.5, 5, 1, 0.5, 5, 1
5010  DATA 0.04, 8, 3.3, 0
```

则得到闭环特征方程不含纯滞后项的闭环输出响应，如图 4－33 所示。

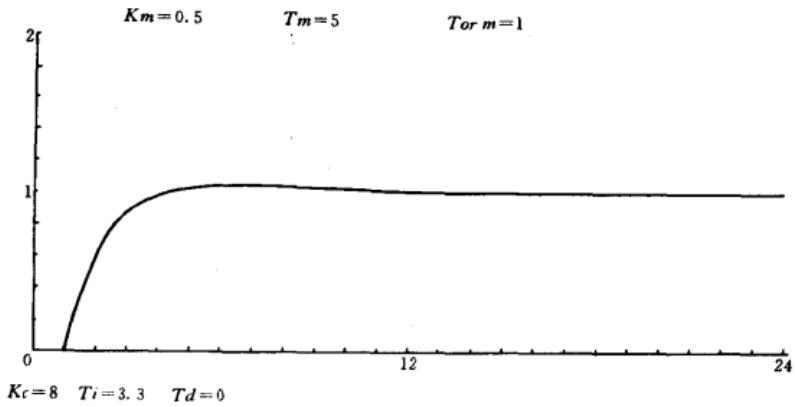


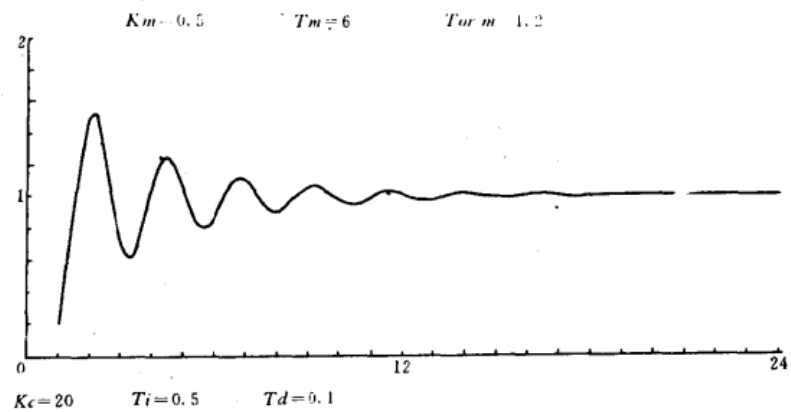
图 4－33　完全补偿时的闭环输出

而上述的 *PI* 参数若应用于对象，组成的闭环系统与图 2－17 相同。其响应示于图 2－20。

不完全补偿时，设 $K_m=0.5$，$T_m=6$，$\tau_m=1.2$，输入数据语句成为：

5000　DATA 0.5，5，1，0.5，6，1.2

5010　DATA 0.04，20，0.5，0.1



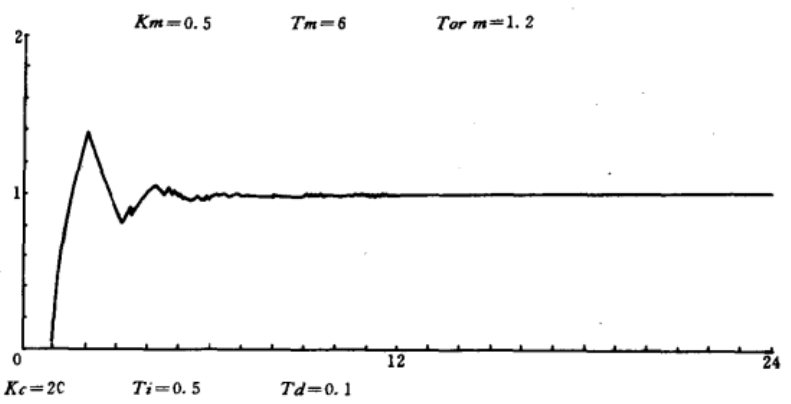


图 4－34　不完全补偿时的闭环输出

比较 *PID* 控制与预测 *PID* 控制的闭环输出。采用相同的参数：$K_c=20$；$T_i=0.5$，T_d

=0.1。其结果示于图 4－33（a）及图 4－33（b）。

有时，为了比较各控制算法或控制方案的优劣，需计算控制指标，常用的有 ISE，ITAE 等。通过简单地加入若干语句到本程序中，即可求取相应的积分指标的值。当然，从图 4－33 可以定性地对 *PID* 控制算法和预测 *PID* 算法进行比较。而从图 4－33 与图 4－32 则可看出史密斯预估补偿控制方案在补偿不完全时特性变差。

读者可以根据上面介绍的数字仿真程序，举一反三，编制观测补偿器控制系统或解析预估补偿控制系统的数字仿真程序。

5 连续系统频域分析法的数字仿真

众所周知，分析控制系统的性能，除了采用时域法测取控制系统的时域响应外，还可采用频域分析的方法对系统进行分析和研究。由于频域法具有下列优点，不仅在经典控制理论中占有很重要的地位，而且在现代控制理论中得到了发展。

①系统的频率特性能用实验方法直接测出，其频率响应的物理意义比较直观。

②能应用于高阶的和具有纯滞后的系统，也可用于某些非线性系统。

③不必求解闭环特征方程，可用图解法对系统进行分析。

④应用于多变量系统时，也同样有效和方便。

频率特性的常用表示方法有极坐标图、对数坐标图和对数幅相图。由于所采用的开环频率特性是等幅振荡时的频率特性，所以求得的闭环特性也是等幅振荡的。如果能用某一衰减比的衰减频率特性曲线进行计算，那么求得的闭环特性也将是该衰减比衰减的。实际参数整定时，若能直接得到某一衰减比的闭环特性，可以简化分析的过程，因此，本章除了介绍极坐标图、对数坐标图外，还给出了衰减频率特性的数字仿真程序。

作为控制系统稳定性的判据，劳斯（Routh）判据是最简单的一种，虽然它是一种代数判据，但由于它简单实用，可直接判据系统是否稳定，因此得到了广泛的应用。为此，本章还介绍了劳斯判据的有关数字仿真程序。

作为经典频率域方法的推广，由卢森布劳克（Rosenbrock）提出的逆奈氏阵列法在多变量控制系统中得到了发展，本章最后对逆奈氏阵列法设计控制器的方法进行了介绍。

5.1 频 率 特 性

5.1.1 典型开环传递函数及其频率特性

5.1.1.1 频率特性

对于一个传递函数为 $G(s)$的线性系统，若其输入变量 $x(t)$是一个角频率为 ω 的正弦波信号，即

$$x(t) = X\sin\omega t$$

则在该系统达到定常状态时，系统的输出 $y(t)$也是一个角频率为 ω 的正弦波，它仅在幅值和相位上发生变化

$$y(t) = Y\sin(\omega t + \varphi)$$

系统的输出和输入的幅值之比用 A 表示

$$A = \frac{Y}{X}$$

系统输出与输入的相位差用 φ 表示。它们都是 ω 的函数。在一定频率范围内进行测试，可以得到一系列的幅值比 A 与相位差 φ 的值，从而得到该系统 $G(s)$ 的频率特性。

在已知传递函数 $G(s)$ 的情况下，只要用 $s = j\omega$ 代入 $G(s)$，得到复向量 $G(j\omega)$，则 $G(j\omega)$ 的模值 $|G(j\omega)|$ 就是幅值比 A，相位角 $\angle G(j\omega)$ 就是相位差 φ。

若 $G(j\omega)$ 的实部数值用 Re 表示，虚部数值用 Im 表示，则

$$|G(j\omega)| = \sqrt{Re^2 + Im^2}$$
$$\underline{/G(j\omega)} = \tan^{-1}(Im/Re)$$

由于 $\tan^{-1}(Im/Re)$ 是取反正切函数的主值范围，因此，在实际计算时，要注意 $G(j\omega)$ 所在的象限，并确定相位角。

用直角坐标系来表示 $G(j\omega)$ 的实频 $Re(\omega)$ 与虚频 $Im(\omega)$ 的关系，得到极坐标图，又称奈奎斯特(Nyquist)图。

当 $G(j\omega)$ 是分式形式表示时，为了求出 $|G(j\omega)|$，应分别求出分子项和分母项的模值，然后相除。同样，当 $G(j\omega)$ 是若干项串联组成时，为了求出 $|G(j\omega)|$，又必须分别求出各项的模值，然后相乘。采用对数可以把相乘、相除的运算简化为相加和相减的形式，因此，用 $\lg G$ 为纵坐标，用 $\lg\omega$ 为横坐标，就得到对数幅频图，同样用 φ 为纵坐标，用 $\lg\omega$ 为横坐标，就得到对数相频图，这二种图统称对数坐标图，亦称伯德 (Bode) 图。由于幅值采用对数标尺，所以环节的添加对合成系统的影响可方便地获得。由于频率 ω 采用对数标尺，所以可以在很宽的频率范围内对系统特性进行研究，使系统在低频段仍有较高精度。

如用 $\lg|G(j\omega)|$ 为纵坐标，用 φ 为横坐标，则得到对数幅相图，亦称尼柯尔斯 (Nichols) 图。它在用 M 轨线法分析闭环频率特性时经常应用。

在多变量控制系统中，常采用倒频率特性来分析系统。若系统的传递函数矩阵是 $G(s)$，则它的逆矩阵 $G_{(s)}^{-1}$ 的频率特性称为倒频率特性。

衰减频率特性，又称扩充频率特性，当衰减度 $m=0$ 时，就可得到上述的频率特性。这部分内容将在下一节介绍。

5.1.1.2 典型环节的频率特性

考虑到通用性和灵活性，典型的环节可分为 S^n、$(1+Ts)$、$(1+as+bs^2)$ 及 $e^{-\tau s}$ 四类。其频率特性分述如下：

①S^n 的频率特性：

$G(s)=S^n$，n 是整数。则

$$G(j\omega) = (j\omega)^n$$
$$\therefore \quad |G(j\omega)| = \omega^n \tag{5-1}$$
$$\underline{/G(j\omega)} = \frac{n}{2}\pi = 90^\circ \times n \tag{5-2}$$

②$1+TS$ 的频率特性：

$$G(s) = 1 + Ts$$
$$\therefore G(j\omega) = 1 + jT\omega;\ Re = 1;\ Im = T\omega$$
$$\therefore |G(j\omega)| = \sqrt{1+T^2\omega^2} \tag{5-3}$$
$$\underline{/G(j\omega)} = \underline{/1+j\omega T} \tag{5-4}$$

当 $T\omega<0$，$\underline{/G(j\omega)} = 180 + 57.29578ATN(T\omega)$

当 $T\omega>0$，$\underline{/G(j\omega)} = 57.29578ATN(T\omega)$

ATN(·)表示 $\tan^{-1}(\cdot)$。

③$1+as+bs^2$ 的频率特性：

$\because G(s)=1+as+bs^2$

$\therefore G(j\omega)=1+ja\omega+b(j\omega)^2=(1-b\omega^2)+ja\omega$

$Re=1-b\omega^2;\ Im=a\omega$

$$\therefore |G(j\omega)|=\sqrt{(1-b\omega^2)^2+a^2\omega^2} \tag{5-5}$$

$$\underline{/G(j\omega)}=\underline{/(1-b\omega^2)+ja\omega} \tag{5-6}$$

当$(1-b\omega^2)<0$时，$\underline{/G(j\omega)}=180+57.29578\mathrm{ATN}[a\omega/(1-b\omega^2)]$

当$(1-b\omega^2)\geqslant 0$时，$\underline{/G(j\omega)}=57.29578\mathrm{ATN}[a\omega/(1-b\omega^2)]$

④$e^{-\tau s}$的频率特性：

$\because G(j\omega)=e^{-\tau j\omega}$

$$\therefore |G(j\omega)|=1 \tag{5-7}$$

$$\underline{/G(j\omega)}=-57.29578\tau\omega \tag{5-8}$$

5.1.2 频率特性的分析方法

5.1.2.1 稳定性在开环频率特性上的体现

从频率域看，若在某一频率时，系统的闭环特性的幅值比为无穷大，那么由于一个连续函数可以分解为一系列的谐波的叠加，则使输入信号会含有该频率，从而使输出成为无穷大。这表明该系统处于稳定的边缘。要使

$$\left|\frac{G(j\omega)}{1+G(j\omega)}\right|\to\infty$$

表明闭环系统达到稳定边缘的充要条件是：

$$1+G(j\omega)=0$$

即 $G(j\omega)=-1+j0$，ω 有实数解。 (5-9)

上式表明开环频率特性 $G(j\omega)$ 通过 $(-1,j0)$ 点时，相应闭环系统处于稳定边缘。

我们知道，闭环特征方程 $1+G(s)=0$ 的根若位于 s 复平面的虚轴上，则闭环系统处于稳定边缘。用 $s=j\omega$ 代入 $G(s)$ 得到的频率特性 $G(j\omega)$ 可以理解为 s 复平面上的虚轴在 $G(s)$ 复平面的映射。见图 5-1。

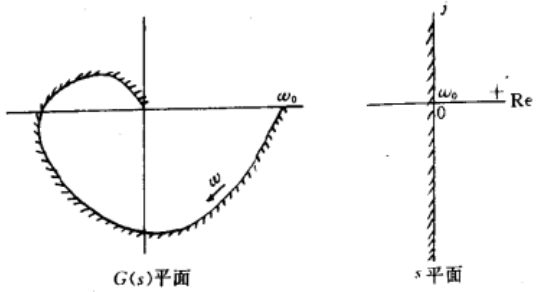


图 5-1 s 平面在 $G(s)$ 平面的映射

可以看到，s 复平面的虚轴上的点对应于 $G(j\omega)$ 的相应点。如 $G(j\omega)$ 通过 $(-1,j0)$ 点，则表示在该点上有 $1+G(j\omega)=0$ 成立，则系统是处于稳定边缘。如果 $G(j\omega)$ 曲线左侧划上阴影线，那么如 $(-1,j0)$ 在该区域内，则系统就是稳定的，换句话讲，若 $G(j\omega)$ 的曲线在与负实轴交点处的幅值小于 1，则该系统就是稳定的。

当 ω 为负时画出的频率特性曲线 $G(-j\omega)$ 与 ω 为正时的频率特性曲线 $G(j\omega)$ 在极坐标图上是上下对称的。因此，通常只讨论 ω 为正频率特性曲线。

5.1.2.2 稳定裕度

在频域分析中，系统稳定性的衡量常用稳定裕度。图 5-2 分别说明了稳定裕度在极坐标图和对数坐标图的定义。

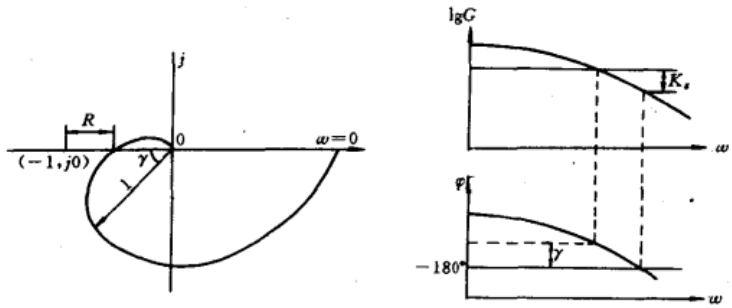


图 5－2　稳定裕度

在 $\varphi=-180°$ 时，$|G(j\omega)|$ 小于1的差值称为幅稳定裕度。在极坐标图上，定义 $R=1-|G(j\omega)|$ 作为幅稳定裕度。在对数坐标图上，定义

$$K_g=-20\lg|G(j\omega)|=20\lg\frac{1}{|G(j\omega)|}$$

即该点分贝数的负值作为幅稳定裕度。

在 $|G(j\omega)|=1$ 的单位圆与 $G(j\omega)$ 曲线的交点处，相位角 φ 与 $-180°$ 的差值称为相稳定裕度。定义 $\gamma=\varphi-(-180°)=\varphi+180°$ 作为相稳定裕度。

应该指出，上述稳定裕度的度量对于开环稳定的系统，是正确的。对于开环不稳定的系统，只有 $G(j\omega)$ 正向穿越 $(-1,j0)$ 点以左的区域，闭环系统才能稳定。因此，R 与 γ 都为正值时，闭环系统是不稳定的。在应用时要注意。

按一般经验准则，常取 $R=0.5$ 左右，γ 取 30°～60°间。

当闭环系统可用两阶振荡环节近似时，即

$$\frac{G(s)}{1+G(s)}=\frac{\omega_0^2}{1+2\zeta\omega_0 s+\omega_0^2}$$

可以推得开环传递函数

$$G(s)=\frac{\omega_0^2}{s(s+2\zeta\omega_0)}$$

当 $\varphi=-180°$ 时，$\omega\to\infty$，$|G(j\omega)|=0$，即 $R=1$，这表明 R 与 ζ 无关。

当 $|G(j\omega)|=1$ 时，有

$$\left|\frac{\omega_0^2}{j\omega(j\omega+2\zeta\omega_0)}\right|=1$$

即：$\omega=\omega_0\sqrt{\sqrt{1+4\zeta^2}-2\zeta^2}$

$$\varphi=\angle\left(\frac{1}{j\omega}\right)+\angle\left(\frac{1}{j\omega+2\zeta\omega_0}\right)=-90°\tan^{-1}\left(\sqrt{\frac{\sqrt{1+4\zeta^2}-2\zeta^2}{2\zeta}}\right)$$

$$\gamma=180°+\varphi=90°-\tan^{-1}\left(\sqrt{\frac{\sqrt{1+4\zeta^2}-2\zeta^2}{2\zeta}}\right)$$

$$=\tan^{-1}\left(\frac{2\zeta}{\sqrt{\sqrt{1+4\zeta^2}-2\zeta^2}}\right)$$

当 $\gamma=30°$ 求得 $\zeta=0.26$；

当 $\gamma=60°$ 求得 $\zeta=0.60$。

5.1.2.3　闭环频率特性和性能指标

工程应用中，常把闭环系统作为二阶振荡环节近似，并据此类推到实际系统。这种近似虽然有误差，如不适用于有余差的系统，闭环特性发生变化，但作为工程应用是允许的。

若闭环系统的频率特性是$\frac{G(s)}{1+G(s)}$，它的幅值比的最大值用M_m表示，相应的频率称共振频率，用ω_R表示，则闭环系统近似为二阶振荡环节时，有下列关系式成立。

①阻尼系数ζ与M_m的关系：

$$M_m = \frac{1}{2\zeta\sqrt{1-\zeta^2}} \tag{5-10}$$

②超调量与M_m的关系：

$$\text{超调量} = \exp\left(-\zeta\pi/\sqrt{1-\zeta^2}\right) \tag{5-11}$$

③工作频率与共振频率的关系：

$$\text{工作频率}\ \omega = \omega_0\sqrt{1-\zeta^2} \tag{5-12}$$

$$\text{共振频率}\ \omega_R = \omega_0\sqrt{1-2\zeta^2} \tag{5-13}$$

④衰减度m、衰减比n与阻尼系数的关系：

$$m = \frac{\zeta}{\sqrt{1-\zeta^2}};\ n = \exp(2\pi m) = \exp\left(\frac{2\pi\zeta}{\sqrt{1-\zeta^2}}\right) \tag{5-14}$$

下列数据说明了它们与M_m的关系：

衰减比	衰减度	阻尼系数	M_m
4:1	0.221	0.216	2.37
10:1	0.366	0.343	1.48

5.1.3　控制器参数整定

5.1.3.1　纯比例调节规律

若开环系统传递函数是$G(s)$，采用纯比例控制器，则根据稳定裕度的经验准则，有二种方法确定控制器的增益K_c。

①求出$G(j\omega)$在临界频率ω_c时的幅值$|G(j\omega_c)|$。则K_c根据

$$K_c = \frac{0.5}{|G(j\omega_c)|}$$

求得。它是满足幅稳定裕度条件的解。

②按所需的相稳定裕度值γ，作$\varphi = -180° + \gamma$的直线与$G(j\omega)$相交，交点的频率设为ω_g，则K_c由

$$K_c = \frac{1}{|G(j\omega_g)|}$$

求得。

5.1.3.2　比例积分调节规律

比例积分调节规律要确定两个参数K_c和T_i。一般T_i按经验准则取$T_i = (0.8\sim1)T_c$，其中T_c是纯比例作用下过程的临界振荡周期。然后用$G'(s) = K_c\left(1+\frac{1}{T_i s}\right)G(s)$作为新的开环系统，按5.1.3.1的方法确定$K_c$。通过$T_i$的调试，可以使采用上述两种方法求得的$K_c$

比较接近，从而使控制系统既能有合适的幅稳定裕度，又能够有足够的相稳定裕度。

5.1.3.3 比例积分微分调节规律

与5.1.3.2相类似，积分时间 T_i 和微分时间 T_d 常常采用经验准则来确定。

$$T_i = (0.3 \sim 0.6)T_c; T_d = \left(\frac{1}{4} \sim \frac{1}{3}\right) T_i$$

然后按5.1.3.1的方法确定 K_c。应该指出，这时的开环系统是 $K_c\left(1+\frac{1}{T_i s}+T_d s\right)G(s)$。

关于 T_c 的确定方法，简述如下，作出 $G(j\omega)$ 的频率特性，求出当 $\varphi = \angle G(j\omega) = -180°$ 时的频率，设为 ω_c，这个频率就是纯比例作用下过程的临界振荡频率。则 $T_c = \frac{2\pi}{\omega_c}$。

5.2 衰减频率特性

5.2.1 衰减频率特性

在上节的讨论中，我们知道，若系统的输入信号是正弦波，则系统达到定常状态时，系统输出也是相同频率的正弦波。频率特性就是系统输出和输入的幅值比和相位差随频率变化的函数关系。根据相同的道理，如果输入衰减振荡波信号。即

$$x(t) = Xe^{-m\omega t}\sin\omega t$$

则稳定的输出信号也是相同频率的衰减振荡波。即

$$y(t) = Ye^{-m\omega t}\sin(\omega t - \varphi)$$

按相同的方法，可以求出幅值比 A 和相位差 φ，并得到衰减频率特性。

〔例5-1〕 已知 $G(s) = \frac{K}{Ts+1}$，求其衰减频率特性。

开环系统的微分方程是

$$T\frac{dy}{dt} + y = Kx$$

$\because x(t) = Xe^{-m\omega t}\sin\omega t$。$m$ 称为衰减度。写成指数形式是：

$$x(t) = Xe^{-m\omega t}e^{j\omega t} = Xe^{(j-m)\omega t}$$

同样，输出 $y(t) = Ye^{-m\omega t}\sin(\omega t - \varphi)$ 也写成指数形式是：

$$y(t) = Ye^{-m\omega t}e^{j(\omega t-\varphi)} = Ye^{(j-m)\omega t}e^{-j\varphi}$$

$$\therefore \quad \frac{dy(t)}{dt} = Ye^{-j\varphi}(j-m)\omega e^{(j-m)\omega t}$$

代入微分方程，有

$$TYe^{-j\varphi}(j-m)\omega e^{(j-m)\omega t} + Ye^{(j-m)\omega t}e^{-j\varphi} = KXe^{-m\omega t}e^{j\omega t}$$

$$\therefore \quad Ye^{-j\varphi}[T(j-m)\omega + 1] = KX$$

即

$$Ye^{-j\varphi} = \frac{KX}{T(j-m)\omega + 1}$$

用 $G(j\omega, m)$ 表示衰减频率特性，则有

$$G(j\omega, m) = \frac{K}{T(j-m)\omega + 1}$$

由此可见，在求频率特性时，只须将 $s = j\omega$ 代入系统传递函数 $G(s)$，而求衰减频率特性时，

是用 $s=(j-m)\omega$ 代入 $G(s)$。如前所述，频率特性是 s 复平面的虚轴 $s=j\omega$ 映射到 $G(s)$ 平面，而衰减频率特性是 s 复平面的等衰减比线 $s=(j-m)\omega$ 映射到 $G(s)$ 平面。图 5-3 表示了这种映射关系。

5.2.2 衰减比判据

奈奎斯特稳定性判据指出，系统的开环频率特性通过（-1，$j0$）点时，相应的闭环系统处于稳定的边缘。这时有

$$1+G(j\omega)=0 \quad 或 \quad G(j\omega)=-1=e^{-j\pi}$$

当（-1，$j0$）点在 $G(j\omega)$ 曲线的左侧，则闭环系统是稳定的，当（-1，$j0$）点在 $G(j\omega)$ 曲线的右侧，则闭环系统是不稳定的。

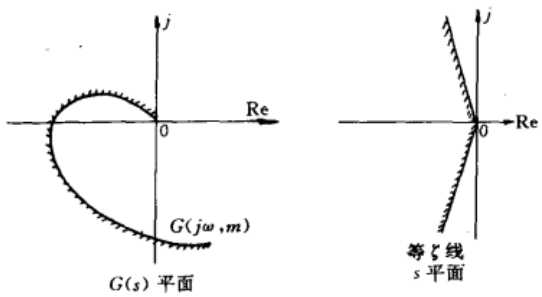


图 5-3 衰减频率特性

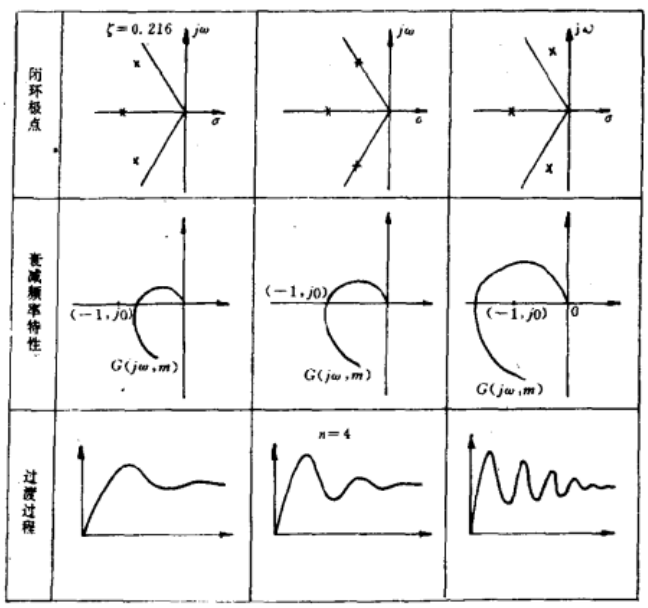


图 5-4 衰减振荡边界条件

在衰减频率特性的情况下，奈奎斯特判据推广为衰减比判据，即当衰减频率特性 $G(j\omega, m)$ 与 $(-1, j0)$ 点相交时，系统处于一个具有一定衰减比的边界状态。例如，如按 $m=0.221$ 绘制的 $G(j\omega, m)$ 曲线与 $(-1, j0)$ 相交，表明闭环系统处于 4:1 的衰减振荡状态。同样，当 $(-1, j0)$ 点在 $G(j\omega, m)$ 曲线的左侧，表明闭环系统的衰减比大于 m 所对应的衰减比 n，如例中 n 为 4:1。当 $(-1, j0)$ 点在 $G(j\omega, m)$ 曲线的右侧，表明闭环系统的衰减 比小于 m 所对应的衰减比 n。图 5-4 是衰减振荡边界条件的示意图。

5.2.3 用衰减频率特性整定控制器参数

衰减比通常是一般工业过程控制系统的主要性能指标，按稳定裕度法整定控制器参数时，系统能否达到规定的衰减比是没有把握的。从所需的衰减比出发进行参数整定要比用稳定裕度指标更切合实际需要。由于计算机的应用，使得绘制衰减频率特性变得十分方便，这使得用衰减频率法整定控制器参数变得更实用。同时，由于当 $m=0$ 时，可以得到频率特性，因此，可以很容易对采用衰减频率特性整定的参数进行稳定性的校验。

5.2.3.1 纯比例调节规律

若开环系统传递函数是 $G(s)$，作出所需衰减度 m 时的衰减频率特性曲线 $G(j\omega, m)$，设曲线与负实轴的交点处频率为 ω_{cn}，则 K_c 可根据

$$K_c = \frac{1}{|G(j\omega_{cn}, m)|}$$

求得。ω_{cn} 称为纯比例作用时的工作频率。

5.2.3.2 比例积分调节规律

作出衰减频率特性曲线 $G(j\omega, m)$，并求得 ω_{cn}，则工作周期 $T_{cn}=\frac{2\pi}{\omega_{cn}}$ 可求得，按经验准则，用 $T_i=\left(\frac{1}{2}\sim 1\right)T_{cn}$ 整定值作为积分时间，然后按新的开环系统 $G'(s)=\left(1+\frac{1}{T_i s}\right)G(s)$ 作出衰减频率特性曲线，则

$$K_c = \frac{1}{|G'(j\omega, m)|}$$

作为稳定性校验，可对 $K_c G'(s)$ 用 $m=0$ 作衰减频率特性曲线，则相稳定裕度和幅稳定裕度立刻可以得到。

5.2.3.3 比例积分微分调节规律

如5.2.3.2所述，先求出 T_{cn}，然后按经验准则，用 $T_i=(0.3\sim 0.6)T_{cn}$，$T_d=\left(\frac{1}{4}\sim\frac{1}{3}\right)T_i$ 设置积分时间和微分时间，对新的开环系统 $G'(s)=\left(1+\frac{1}{T_i s}+T_d s\right)G(s)$ 作出衰减频率特性曲线，按 $K_c=\frac{1}{|G'(j\omega, m)|}$ 整定比例部分的增益。同样，当 $m=0$ 时，以 $K_c G'(s)$ 为对象作出的衰减频率特性曲线能够用来进行闭环系统稳定裕度的校验。

5.2.3.4 实例

〔例5-2〕 某开环系统 $G(s)=\frac{1}{(1+5s)^2(1+2s+4s^2)}$，采用比例积分调节规律，若选用 $T_i=1$，确定4:1衰减比时的控制器放大倍数 K_c。

作出 $K_c=1$ 时的广义对象 $G'(s)=\frac{1}{(1+5s)^2(1+2s+4s^2)}\cdot\frac{1+s}{s}$ 的衰减频率特性，如下所示。

w	Re	Im	G	P
.1	−9.582	−2.293	9.852	−166.541
.102	−9.368	−2.001	9.58	−167.944
.105	−9.152001	−1.717	9.312	−169.371
.107	−8.932999	−1.443	9.049	−170.822
.11	−8.71	−1.178	8.79	−172.297
.112	−8.486	−.923	8.536	−173.795
.115	−8.258001	−.676	8.286	−175.317
.117	−8.028	−.44	8.04	−176.862
.12	−7.797	−.214	7.8	−178.43
.123	−7.563	.003	7.563	−180.021
.126	−7.328	.209	7.331	−181.634

得到 $P=-180°$ 时的 $|G'(j\omega, m)|=7.563$；$\omega_{cn}=0.123$；因此有 $K_c=\frac{1}{|G'(j\omega, m)|}=0.132$。以此值作出 $m=0$ 的 $G'(j\omega, m)$，可用来求得闭环系统的稳定裕度。当 $m=0$，$K_c=0.132$ 时的衰减频率特性如下。

w	Re	Im	G	P
.078	−1.098	−1.023	1.5	−137.025
7.900001E−02	−1.084	−.975	1.458	−138.037
.081	−1.07	−.928	1.417	−139.068
.083	−1.056	−.883	1.376	−140.117
.085	−1.042	−.838	1.337	−141.185
.087	−1.027	−.794	1.298	−142.271
.089	−1.011	−.752	1.26	−143.376
.091	−.996	−.71	1.223	−144.5
.093	−.98	−.67	1.187	−145.644
.095	−.963	−.63	1.151	−146.806
.098	−.947	−.592	1.116	−147.987
.1	−.929	−.554	1.082	−149.188
.102	−.912	−.518	1.049	−150.408
.105	−.894	−.483	1.016	−151.647
.107	−.876	−.448	.984	−152.905
.11	−.858	−.415	.953	−154.182
.112	−.84	−.383	.923	−155.479
.115	−.821	−.352	.893	−156.795
.117	−.802	−.322	.864	−158.13
.12	−.782	−.293	.835	−159.484
.123	−.763	−.265	.808	−160.857
.126	−.743	−.238	.78	−162.249
.129	−.723	−.212	.754	−163.66

.132	-.704	-.187	.728	-165.09
.135	-.683	-.164	.703	-166.539
.138	-.663	-.141	.678	-168.006
.141	-.643	-.119	.654	-169.492
.145	-.623	-.099	.631	-170.996
.148	-.603	-7.900001E-02	.608	-172.518
.151	-.583	-.061	.586	-174.059
.155	-.562	-.043	.564	-175.618
.158	-.542	-.027	.543	-177.196
.162	-.523	-.011	.523	-178.791
.166	-.5030001	.004	.5030001	-180.405
.17	-.483	.017	.483	-182.036
.174	-.464	.03	.465	-183.686
.178	-.444	.042	.446	-185.354
.182	-.426	.053	.429	-187.04
.186	-.407	.063	.412	-188.745
.191	-.388	.072	.395	-190.468

可以看到，$P=-180°$时，$|G'(j\omega,m)|=0.503$，因此，幅稳定裕度 $R=1-|G'(j\omega,m)|=0.497$。同样，$|G'(j\omega,m)|=1$ 时，$P=-152°$，因此，相稳定裕度 $\gamma=180°+P=28°$。

〔例 5-3〕某开环系统 $G(s)=\dfrac{1}{(1+5s)^2(1+2s+4s^2)}$，采用比例积分调节规律，确定 4:1 衰减比时的控制器放大倍数 K_c 及积分时间 T_i。

作出 $K_c=1$ 时 $G(s)$ 的衰减频率特性，如下所示。

w	Re	Im	G	P
.302	-.534	-.067	.538	-172.818
.309	-.526	-.039	.527	-175.767
.316	-.5170001	-.011	.5170001	-178.794
.324	-.5070001	.017	.5070001	-181.904
.331	-.495	.044	.497	-185.109
.339	-.482	.071	.487	-188.416
.347	-.467	.098	.477	-191.836
.355	-.451	.124	.468	-195.381
.363	-.433	.15	.458	-199.063
.372	-.413	.174	.448	-202.892

求得 $P=-180°$时的 $\omega_{cn}=0.32$；

$\therefore T_{cn}=\dfrac{2\pi}{0.32}=19.63$；取 $T_i=T_{cn}=19.63$；

广义对象为

$$G'(s)=\frac{(9.8s+1)}{s}\cdot\frac{1}{19.63}\cdot K_c\left(\frac{1}{(1+5s)^2(1+2s+4s^2)}\right)$$

作出 $K_c=1$ 时$G'(s)$的衰减频率特性，如下示。

w	Re	Im	G	P
.288	−.546	−.028	.547	−177.038
.289	−.545	−.026	.546	−177.294
.29	−.544	−.023	.545	−177.551
.29	−.543	−.021	.544	−177.808
.291	−.542	−.018	.543	−178.066
.292	−.541	−.016	.541	−178.324
.292	−.54	−.013	.54	−178.583
.293	−.539	−.011	.539	−178.843
.294	−.538	−8.000001E−03	.538	−179.104
.294	−.537	−.006	.537	−179.365
.295	−.536	−.003	.536	−179.627
.296	−.535	−.001	.535	−179.889
.296	−.534	.001	.534	−180.152
.297	−.533	.004	.533	−180.416
.298	−.532	.006	.532	−180.68
.299	−.531	8.999999E−03	.531	−180.946
.299	−.53	.011	.53	−181.212
.3	−.529	.014	.529	−181.478

得 $K_c=\frac{1}{0.534}$；$\omega_{cn}=0.296$。

以 $m=0$ 代入，可以得到频率特性如下：

w	Re	Im	G	P
.207	−.676	−.747	1.007	−132.171
.207	−.677	−.743	1.005	−132.347
.207	−.678	−.739	1.003	−132.523
.208	−.679	−.735	1.001	−132.699
.208	−.679	−.732	.998	−132.875
.209	−.68	−.728	.996	−133.052
.209	−.681	−.724	.994	−133.23
.21	−.682	−.721	.992	−133.407
.21	−.682	−.717	.99	−133.585
.339	−.564	−.011	.564	−178.859
.34	−.562	−8.999999E−03	.562	−179.129
.34	−.561	−.006	.561	−179.4
.341	−.559	−.003	.559	−179.671
.342	−.557	−.001	.557	−179.943
.343	−.555	.002	.555	−180.215
.344	−.553	.005	.553	−180.488
.344	−.551	.007	.551	−180.762
.345	−.549	.01	.549	−181.036
.346	−.547	.013	.547	−181.311

由此求得幅稳定裕度 $R=1-0.555=0.445$。相稳定裕度 $\gamma=180^\circ-132.7^\circ=47.3^\circ$。

例 5−2 与例 5−3 表明，T_i 减小，ω_{cn}降低，稳定程度下降。通过调整 K_c，可以得到相同的衰减比，但调节过程变差。

5.2.4 衰减频率特性仿真程序

衰减频率特性仿真程序适用于开环传递函数形如：

$$G(s)=\frac{K(1+z_1s)(1+z_2s)\cdots(1+z_{m_1}s)(1+z_{11}s+z_{12}s^2)(1+z_{21}s+z_{22}s^2)}{(1+p_1s)(1+p_2s)\cdots(1+p_{n_1}s)(1+p_{11}s+p_{12}s^2)(1+p_{21}s+p_{22}s^2)}\rightarrow$$

$$\leftarrow\frac{\cdots(1+z_{m_21}s+z_{m_22}s^2)s^{m3}e^{-DT\cdot s}}{\cdots(1+p_{n_21}s+p_{n_22}s^2)} \tag{5-15}$$

对分子项与分母项分别计算幅值与相位角，然后，用分子项幅值除以分母项幅值，得到总的幅值。用分子项相位角减去分母项相位角，得到总的相位角。

5.2.4.1 形如（1+Bs）的环节衰减频率特性的计算

若 $G(s)=1+Bs$，则用 $s=j\omega-m\omega$ 代入得到相应的实频和虚频特性是：

$$RE=1-m\omega B;IM=\omega B$$

因此，幅值与相位角可由下式计算：

$$|G(j\omega,m)|=\sqrt{RE^2+IM^2};\underline{/G(j\omega,m)}=\tan^{-1}\left(\frac{IM}{RE}\right)+\frac{\pi}{2}\times D; \tag{5-16}$$

$$D=\mathrm{sgn}(IM)*(1-\mathrm{sgn}(RE)) \tag{5-17}$$

5.2.4.2 形如 $(1+Fs+Vs^2)$ 环节的衰减频率特性的计算

若 $G(s)=(1+Fs+Vs^2)$，用 $s=(j-m)\omega$ 代入，得到实频与虚频特性。

$$RE=1-Fm\omega-V(m^2-1)\omega^2;IM=F\omega-2mV\omega^2$$

按（5-16）与（5-17）同样可以求得相应的幅值与相位角。

5.2.4.3 形如 s^m 环节的衰减频率特性的计算

若 $G(s)=s^m$，其中 m 为正，表示在分子项，m 为负表示在分母项。用 $s=(j-m)\omega$ 代入得相应的幅值与相位角：

$$|G(j\omega,m)|=\sqrt{[(1+m^2)\omega^2]^m};\underline{/G(j\omega,m)}=\left[180-57.29578\tan^{-1}\left(\frac{1}{m}\right)\right]^m;$$

当 $m=0$ 时，$\underline{/G(j\omega,m)}=90\times m$；

环节的实频与虚频分别是：

$$RE=-m\omega;IM=\omega;$$

5.2.4.4 形如 $e^{-\tau s}$ 环节的衰减频率特性的计算

若 $G(s)=e^{-\tau s}$，用 $s=(j-m)\omega$ 代入，得实频与虚频特性：

$$RE=e^{m\tau\omega}\times\cos\varphi;IM=e^{m\tau\omega}\times\sin\varphi;$$

$$\varphi=-57.29578\times\tau\times\omega;$$

同样求得幅值与相位角：

$$|G'(j\omega,m)|=e^{m\tau\omega};\underline{/G'(j\omega,m)}=\varphi。$$

应该指出，组成分子项或分母项的各项应幅值相乘，相位角相加，以得到分子项或分母项的总幅值与相位角。放大倍数 K 仅影响分子项的幅值，即乘以 K。

5.2.4.5 衰减频率特性仿真程序

衰减频率特性的仿真程序清单如下：

```
10  REM Decay Frequency Property(Nyquist Method)
15  REM G(s)=P(s)/Q(s)*EXP(-DT*s)
```

```
20  REM P(s)=K(1+z(1)s)*(1+z(2)s)*…*(1+z(m1)s)*(1+zz(1)s+zz(2)s^2)
    *…
25  REM          *(1+zz(2*m2-1)s+zz(2*m2)s^2)*s^m3
30  REM Q(s)=(1+p(1)s)*(1+p(2)s)*…*(1+p(n1)s)*(1+pp(1)s+pp(2)s^2)
    *…
35  REM          *(1+pp(2*n2-1)s+pp(2*n2)s^2)
50  CLS
55  LOCATE 2,1
60  PRINT"Calculate Decay Frequency Property"
65  INPUT"The Number of The Numerator (1+z*s)";M1
70  INPUT"The Number of The Numerator (1+z1*s+z2*s^2)";M2
75  INPUT"The Number of The Numerator s^m3(m3 can+or-)";M3
77  INPUT"The Dead Time of This System";DT
80  INPUT"The Number of The Denominator (1+p*s)";N1
85  INPUT"The Number of The Denominator (1+p1*s+p2*s^2)";N2
90  INPUT"The Degree of Decaym(m=.221=>4:1)";MM
95  INPUT"The Start Frequency w0";W0
100 W0=INT(LOG(W0)/LOG(10)*100+.5)/100
105 INPUT"The Gain of The Open-Loop System";K
110 CLS
115 PRINT"Input The Coeifficient Factors of The Numerator"
120 LPINT"Input The Coeifficient Factors of The Numerator"
125 LPRINT
130 IF M1=0 THEN 160
135 FOR I=1 TO M1
140 PRINT"z(";I;
145 INPUT")=";Z(I)
150 LPRINT"z(";I;")=";Z(I)
155 NEXT I
160 LPRINT
165 IF M2=0 THEN 210
170 FOR I=1 TO 2*M2 STEP 2
175 PRINT"zz(";I;
180 INPUT")=";ZZ(I)
185 LPRINT"zz(";I;")=";ZZ(I)
190 PRINT"zz(";I+1;
195 INPUT")=";ZZ(I+1)
200 LPRINT"zz(";I+1;")=";ZZ(I+1)
205 NEXT I
210 LPRINT
```

```
215  LPRINT "Integrate Item Order:"; -1 * M3:LPRINT
220  CLS
225  PRINT"Input The Coeifficient Factors of The Denominator"
230  LPRINT"Input The Coeifficient Factors of The Denominator"
233  LPRINT
235  IF N1=0 THEN 265
240  FOR I=1 TO N1
245  PRINT"p(";I;
250  INPUT")=";P(I)
255  LPRINT"p(";I;")=";p(I)
260  NEXT I
265  LPRINT
270  IF N2=0 THEN 320
275  FOR I=1 TO 2 * N2 STEP 2
280  PRINT "PP(";I;
285  INPUT")=";PP(I)
290  LPRINT"pp(";I;")=";PP(I)
300  PRINT"pp(";I+1;
305  INPUT")=";PP(I+1)
310  LPRINT"pp(";I+1;")=";PP(I+1)
315  NEXT I:LPRINT
320  LPRINT"The Degree of Decay m=";MM
322  LPRINT:LPRINT"The Gain of The Open-Loop System K=";K
325  CLS
330  PRINT"w   Re      Im      G      P"
335  LPRINT
340  LPRINT"w   Re      Im      G      P"
350  FOR I=1 TO 70
355  PRINT "-";
360  LPRINT"-";
365  NEXT I
370  PRINT
375  LPRINT
380  FOR L=W0 TO W0+3 STEP.01
385  W=10^L
390  G=K * K
395  ANG=0
400  IF M1=0 THEN 435
405  FOR I=1 TO M1
410  B=Z(I)
```

```
415  RE=1-MM*W*B
420  IM=W*B
425  GOSUB 1000
430  NEXT I
435  IF M2=0 THEN 475
440  FOR I=1 TO M2 STEP 2
445  V=ZZ(I+1)
450  F=ZZ(I)
455  RE=1-F*MM*W+V*(MM*MM-1)*W*W
460  IM=F*W-2*MM*V*W*W
465  GOSUB 1000
470  NEXT I
475  GG=G*((1+MM*MM)*W*W)^M3
480  IF MM<>0 THEN AA=ANG+(180-57.29578*ATN(1/MM))*M3
482  IF MM=0 THEN AA=ANG+90*M3
485  ANG=0
490  G=1
500  IF N1=0 THEN 535
505  FOR I=1 TO N1
510  B=P(I)
515  RE=1-MM*W*B
520  IM=W*B
525  GOSUB 1000
530  NEXT I
535  IF N2=0 THEN 575
540  FOR I=1 TO N2 STEP 2
545  V=PP(I+1)
550  F=PP(I)
555  RE=1-F*MM*W+V*(MM*MM-1)*W*W
560  IM=F*W-2*MM*V*W*W
565  GOSUB 1000
570  NEXT I
575  GG=GG/G*(EXP(MM*DT*W))^2
580  ANG=AA-ANG-57.29578*DT*W
585  G=SQR(GG)
590  IF K<0 THEN ANG=ANG+180
595  R=ANG/57.29578
600  A=G*SIN(R)
605  B=G*COS(R)
610  A(2)=INT(A*1000+.5)/1000
```

```
615  A(1) = INT(B * 1000 + .5)/1000
620  A(3) = INT(G * 1000 + .5)/1000
625  A(4) = INT(ANG * 1000 + .5)/1000
630  A(0) = INT(W * 1000 + .5)/1000
635  FOR E=0 TO 4
640  PRINT A(E);SPC(13 - LEN(STRY(A(E))));
645  LPRINT A(E);SPC(13 - LEN(STRY(A(E))));
650  NEXT E
655  IF A(4)< - 182 OR A(3)<.01 THEN 675
660  PRINT
665  LPRINT
670  NEXT L
675  END
1000  G=G * (RE * RE + IM * IM)
1010  IF RE=0 THEN ANG=ANG + SGN(IM) * 90:GOTO 1040
1020  DD=SGN(IM) * (1 - SGN(RE))
1030  ANG=ANG + 90 * DD + 57.29578 * ATN(IM/RE)
1040  RETURN
```

程序中变量说明如下：

M1：分子项$(1+zs)$形式的环节个数。

M2：分子项$(1+z_1s+z_2s^2)$形式的环节个数。

M3：分子项 s 的幂。M3 负值表示在分母项。

DT：分子项纯滞后时间。

N1：分母项$(1+ps)$形式的环节个数。

N2：分母项$(1+p_1s+p_2s^2)$形式的环节个数。

MM：衰减度 m。

W0：频率初值。

K：开环系统放大倍数。

Re：实频。

Im：虚频。

G：幅值比$|G(j\omega,m)|$

P：相位差$\angle G(j\omega,m)$

程序中 380 语句中的步长，现为 0.01，指频率的增量是 $10^{0.01}$。当要求精确计算时，该数值可以改小，如用 0.005。也可以改大，以便确定接近 -180° 相位角或幅值为 1 时的频率，如可用 0.1 等。

程序中的 MM 是衰减度，当 $m=0.221$ 时得到衰减比为 4:1 的衰减频率特性。当 $m=0.366$ 时，得到衰减比为 10:1 的衰减频率特性。由于要进行稳定裕度的校验，所以用 $m=0$ 代入，则可确定稳定裕度，如前面讨论的例子所述。

本程序没有给出极坐标或对数坐标图，这是由于解题时只要了解相位角为 -180° 时的幅值特性及幅值为 1 时的相位特性，如读者感兴趣，可以方便地由已经给出的实频和虚频特性

或幅频和相频特性绘制，或编制有关程序，由计算机绘制。

本程序可以打印出3个十倍频范围内的频率特性，通常以相位角大于－182°或幅值＜0.1为结束依据，当需要更大频率范围内的数据时，只要把380语句中的终止值由W0＋3改为W0＋5，这里的5指5个十倍频。当需要了解全部范围内的频率特性时，也可以删去655语句。

程序运行时，先依次送入M1、M2、M3、DT、N1、N2、MM、W0及K，然后根据M1、M2、N1、N2的值送入分子项及分母项的各环节中的系数。z（I）（$I=1\sim M1$）表示$(1+zs)$中的s项前的系数。ZZ（I）、ZZ（I＋1）为$(1+z_1s+z_2s^2)$中的相应系数。分母项用p代替z，可按相同方法送入系数，则程序会打印出有关信息及频率特性的数据，用上述实例中所介绍的格式打印。

5.3 劳斯判据确定系统稳定性

5.3.1 劳斯稳定判据

稳定性的判别是控制系统分析和设计中的首要问题。线性控制系统的稳定性由系统特征方程的根在复平面上的位置确定。全部特征根在复平面的左半平面，则系统是稳定的。若有位于复平面的右半平面的特征根，则系统是不稳定的。若有根在虚轴，而其余根在复平面的左半平面，则系统是临界稳定的。

如果只需判别系统是否稳定，而不要求知道特征根在复平面上的具体位置，就不必求解特征方程式，直接用方程式系数来进行判别。这种方法就是劳斯判据的判别方法。

对于控制系统稳定性的判别，劳斯判据的方法介绍如下：

设控制系统闭环特征方程$A(s)=0$，即

$$A(s)=a_0s^n+a_1s^{n-1}+\cdots+a_{n-1}s+a_n=0 \tag{5-18}$$

式中，a_0、a_1、…、a_{n-1}、a_n是实数，$a_n\neq0$。

控制系统稳定的必要条件是：

$$a_i>0,i=0,1,\cdots,n-1,n \tag{5-19}$$

当$n\leqslant2$时，（5－19）式还是稳定性的充分条件。当$n>2$时，先列出由闭环特征方程系数组成的劳斯阵列。如下所示。

s^n	a_0	a_2	a_4	a_6	…
s^{n-1}	a_1	a_3	a_5	a_7	…
s^{n-2}	b_1	b_2	b_3	b_4	…
s^{n-3}	c_1	c_2	c_3	c_4	…
⋮	⋮	⋮			
s^2	e_1	e_2			
s^1	f_1				
s^0	g_1				

劳斯阵列中其余元素的计算，按相应的对角元素相乘，然后相减，再除以第一项，即：

$$b_1=\frac{a_1a_2-a_0a_3}{a_1};b_2=\frac{a_1a_4-a_0a_5}{a_1};b_3=\frac{a_1a_6-a_0a_7}{a_1};\cdots$$

$$c_1 = \frac{b_1 a_3 - a_1 b_2}{b_1}; c_2 = \frac{b_1 a_5 - a_1 b_3}{b_1}; \cdots$$

$$d_1 = \frac{c_1 b_2 - b_1 c_2}{c_1}; d_2 = \frac{c_1 b_3 - b_1 c_3}{c_1}; \cdots$$

……

劳斯判据指出，劳斯阵列中第一列元素的符号改变的次数等于该系统闭环特征方程中不稳定根的个数。换句话讲，该系统稳定的充分和必要的条件是劳斯阵列表的第一列元素应全部为正值。

在计算劳斯阵列各元素时，应判别是否存在下列情况：

①若劳斯阵列的某行第一项元素为零，则不论其他项是否有元素或其他元素不全为零，下一行元素将会因除以零而成为不确定。这时，采用一个有限小的正数 ε，例如 $\varepsilon = 0.00001$ 来代替零，并继续下面的元素计算，判别的方法不变。当第一列元素没有符号改变，则表明该系统没有特征根在复平面的右半平面。但因为存在零，说明系统有根在复平面的虚轴上，即系统处于临界稳定状态。

②若劳斯阵列的某一行元素（如第 i 行）全部为零。这时，用它的上一行，即 $i-1$ 行组成辅助多项式，辅助多项式的偶次项系数是 $i-1$ 行的元素，奇次项系数全部为零，多项式的最高阶次是 $n-i+1$。对辅助多项式求导，并用求导后的多项式系数作为第 i 行元素，以代替原来的零。然后继续计算劳斯阵列的其他元素。判别的方法与①所述相同。

5.3.2 劳斯判据确定系统相对稳定性

劳斯判据用于判别闭环系统的特征根是否在复平面的右半平面，它以虚轴为判别的边界线。若对特征方程进行虚轴的平移，则经变换后的特征方程，再用劳斯判据进行判别，则可以确定原特征方程是否有根在平移后的虚轴左边或右边，根据虚轴平移的量，可以确定系统的相对稳定性。

如前所述，衰减频率特性可以用来确定系统闭环是否有符合一定衰减比的极点。虚轴平移后的劳斯判据则可以用来确定系统闭环是否有符合一定过渡时间的极点。

设系统的特征方程是：

$$A(s) = a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n = 0$$

作虚轴平移的变换：

$$s = p + d \tag{5-20}$$

其中 d 是在虚轴的平移量。经平移后得：

$$A(p+d) = a_0 (p+d)^n + a_1 (p+d)^{n-1} + \cdots + a_{n-1} (p+d) + a_n = 0 \tag{5-21}$$

或：

$$B(p) = B_0 p^n + B_1 p^{n-1} + \cdots + B_{n-1} p + B_n = A(p+d) \tag{5-22}$$

式（5-21）也可以表达为：

$$B(p) = a_n + (p+d)(a_{n-1} + (p+d)(a_{n-2} + \cdots + (p+d)a_0)\cdots) \tag{5-23}$$

按我国宋代数学家秦九韶的算法，(5－23) 式可用递推算法表示如下：

$$B_n^{(0)} = a_0 \tag{5-24}$$

$$B_{n-i}^{(k)} = d \cdot B_{n-i}^{(k-1)} + B_{n-i+1}^{(k-1)} \quad (k = 1, \cdots n; i = k, \cdots 1) \tag{5-25}$$

$$B_n^{(k)} = dB_n^{(k-1)} + a_k \quad (k = 1 \cdots, n; i = n, \cdots 0) \tag{5-26}$$

$$B_{n-i} = B_{n-i}^{(n)} \quad (i = 0, \cdots n) \tag{5-27}$$

经虚轴平移后的$B(p)$，则用劳斯判据进行判别后，得到原系统是否在$Re = d$处以右有根。

〔例 5－4〕 已知闭环特征方程是：

$$s^4 + 3s^3 + 4s^2 + 6s + 4 = 0$$

取 $d = -0.1$，得 $s = p - 0.1$ 时的新特征方程

$$(p - 0.1)^4 + 3(p - 0.1)^3 + 4(p - 0.1)^2 + 6(p - 0.1) + 4 = 0$$

写成

$$B(p) = p^4 + 2.6p^3 + 3.16p^2 + 5.286p + 3.4371 = 0$$

然后即可用劳斯判据进行判别。

5.3.3 仿真程序

劳斯判据仿真程序和平移程序都要输入系统的阶数 N，然后以降幂形式送入相应特征方程的系数 a_0、a_1、…、a_n。

劳斯判据的仿真程序在运行后打印出如下结果：

```
s^5 + s^3 + s = 0
Routh Method to Find The Number of Instable Root
s 5  |   1        1      1
s 4  |   0        0      0
s 4  |   5        3      1
s 3  |   .4       .8
s 2  |  -7        1
s 1  |   .8571429
s 0  |   1
The Number of The Instable Root is 2
```

虚轴平移程序仅在 CRT 上显示 $B(p)$ 的有关系数。感兴趣的读者也可以把两个程序联系起来，既完成平移，又完成相对稳定性判据，同时可以不断增加虚轴的平移量（这里指虚轴以左的平移量的值），以了解特征根的分布。

程序中的变量说明如下：

A (I)：闭环特征方程的系数。I=0，1，…，N。以降幂形式排列。

B (I)：经虚轴平移后的特征方程系数。I=0，1，…N。也以降幂形式排列。

N：特征方程的阶次。

X (I)，Y (I)：劳斯阵列的上、下行元素。

s：劳斯阵列中某行第一列元素值为零的标志数。s=0 表示该元素值为 0。

M：不稳定根的个数。在虚轴上的根不包括在内。可通过第一列元素有否为 0 的元素，（程序中显示 0.000001）进行判别。当不稳定根数为 0，而有第一列元素为 0 的情况，则有特征根在虚轴。通过虚轴左移一个很小的平移量，来判别虚轴上根的个数（由不稳定根数确定）。

A1（I）：劳斯阵列计算所得的行元素。

d：平移量。由于 s＝p＋d；即 p＝s－d，因此，向左平移时 d 取负值。通常为了了解稳定性的情况，d 总是取负值的。

劳斯判据仿真程序的框图如图5－5所示。虚轴平移程序比较简单。因此未画出程序框图。

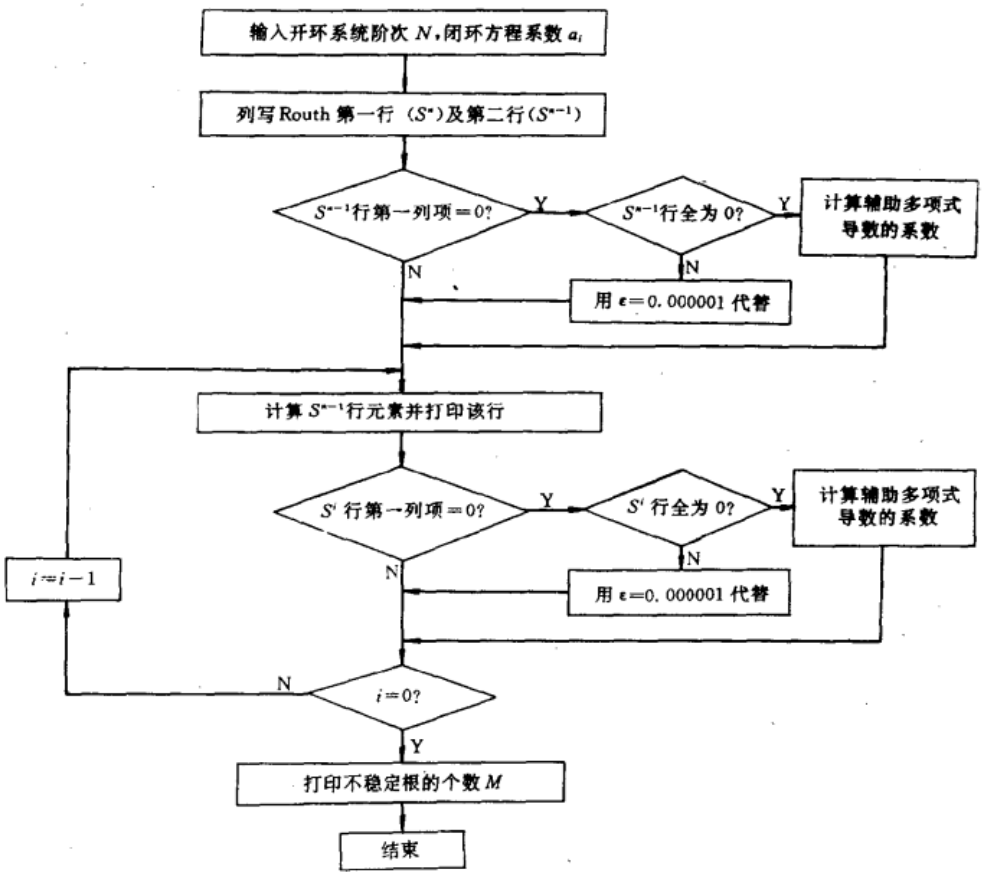


图 5－5　劳斯判据仿真程序框图

劳斯判据仿真程序清单如下：

```
10  REM Decide The Number of Instable Root by Rooth Method
15  REM a(0)s^n + a(1)s^(n - 1) + … + a(n - 1)s + a(n) = 0
20  CLS
25  LOCATE 2, 1
30  INPUT"The Order of The Open - Loop System"; N
35  K = INT((N + 2)/2)
```

```
40  K1=K
45  M=0
50  DIM A1(K),X(K),Y(K),A(N+2)
55  FOR I=1 TO 2*K
60  IF N/2=INT(N/2)AND I=2*K THEN A(I)=0:GOTO 80
65  PRINT"a(";I-1;
70  INPUT")=";A(I)
75  NEXT I
80  FOR I=1 TO N+1
85  IF A(I)=0 THEN 110
90  IF A(I)>0 AND I<>1 THEN LPRINT"+";
95  IF A(I)<>1 OR (A(I)=1 AND I=N+1) THEN LPRINT A(I);
97  IF A(I)=1 THEN LPRINT"  ";
100  IF I<N THEN LPRINT"s";N+1-I;
105  IF I=N THEN LPRINT"s";
110  NEXT I
115  LPRINT"=0"
120  CLS
125  LOCATE 2,1
130  PRINT"Routh Method to Find The Number of Instable Root"
135  LPRINT
140  LPRINT"Routh Method to Find The Number of Instable Root"
145  LPRINT
150  LOCATE 4,1
155  PRINT"s";N;"|  ";
160  LPRINT"s";N;"|  ";
165  S=0
170  FOR I=1 TO K
175  X(I)=A(2*I-1)
180  PRINT X(I);SPC(10-LEN(STRY(X(I))));
185  LPRINT X(I);SPC(10-LEN(STRY(X(I))));
190  NEXT I
195  PRINT
200  LPRINT
205  PRINT"s";N-1;"|  ";
210  LPRINT"s";N-1;"|  ";
215  FOR I=1 TO K
220  Y(I)=A(2*I)
225  S=S+ABS(Y(I))
230  PRINT Y(I);SPC(10-LEN(STRY(Y(I))));
```

```
235  LPRINT Y(I);SPC(10-LEN(STRY(Y(I))));
240  NEXT I
245  PRINT
250  LPRINT
255  IF S<>0 THEN 310
260  PRINT"s";N-1;"|   ";
265  LPRINT "s";N-1;"|   ";
270  FOR I=1 TO K
275  L=2*(I-1)
280  Y(I)=(N-L)*X(I)
285  PRINT Y(I);SPC(10-LEN(STRY(Y(I))));
290  LPRINT Y(I);SPC(10-LEN(STRY(Y(I))));
295  NEXT I
300  PRINT
305  LPRINT
310  IF Y(1)=0 THEN Y(1)=.000001
315  IF Y(1)*X(1)<0 THEN M=M+1
320  FOR I=1 TO N-1
325  S=0
330  K=K1-INT((I-1)/2)
335  IF K=1 THEN K=K+1
340  GOSUB 500
345  NEXT I
350  LPRINT
355  PRINT "The Number of The Instable Root is";M
360  END
500  PRINT"s";N-1-I;"|   ";
505  LPRINT "s";N-1-I;"|   ";
510  FOR L=1 TO K-1
515  A1(L)=(Y(1)*X(L+1)-X(1)*Y(L+1))/Y(1)
520  S=S+ABS(A1(L))
525  PRINT A1(L);SPC(10-LEN(STRY(A1(L))));
530  LPRINT A1(L);SPC(10-LEN(STRY(A1(L))));
535  NEXT L
540  PRINT
545  LPRINT
550  IF N-I+2 OR S<>0 THEN 605
555  PRINT"s";N-2-I;"|   ";
560  LPRINT"s";N-2-I;"|   ";
565  FOR L=1 TO K-1
```

```
570  P=2*(L-1)
575  A1(L)=(N-I-P)*Y(L)
580  PRINT A1(L);SPC(10-LEN(STRY(A1(L))));
585  LPRINT A1(L);SPC(10-LEN(STRY(A1(L))));
590  NEXT L
595  PRINT
600  LPRINT
605  IF A1(1)=0 THEN A1(1)=.000001
610  FOR P=1 TO K
615  X(P)=Y(P)
620  Y(P)=A1(P)
625  NEXT P
630  Y(K)=0
635  IF Y(1)*X(1)<0 THEN M=M+1
640  RETURN
```

虚轴平移程序清单如下：

```
5  CLS
10  REM SHIFT IM AXIS
20  LOCATE 2,1
30  INPUT "Input The Order of The System      ";N
35  DIM A(N),B(N)
40  FOR I=0 TO N
45  LOCATE 4+I*2,1
50  PRINT"A(";I;
55  INPUT")=";A(I)
60  NEXT I
65  PRINT:INPUT "The shift value of Im axis      ";D
70  FOR I=0 TO N-1
75  B(I)=0
80  NEXT I
85  B(N)=A(0)
90  FOR K=1 TO N
95  FOR I=K TO 1 STEP -1
100  B(N-I)=B(N-I)*D+B(N-I+1)
105  NEXT I
110  B(N)=B(N)*D+A(K)
115  NEXT K
120  FOR I=0 TO N
125  PRINT B(I)
130  NEXT I
```

140 END

〔例 5－5〕 用劳斯判据判别系统的稳定性。若系统特征方程是：

$$4s^4 + 3s^3 + 2s^2 + s = 0$$

运行劳斯判据仿真程序。

显示 The Order of The Open－Loop System?

打入 4，并回车。

显示 a（1）＝?

打入 4，并回车。

显示 a（2）＝?

打入 3，并回车

显示 a（3）＝?

打入 2，并回车。

显示 a（4）＝?

打入 1，并回车。

则在打印机及纸张准备好后，会打印下列结果。第一行是闭环特征方程。然后是劳斯阵列，最后是不稳定特征根个数，本例为 0。

4s^4 + 3s^3 + 2s^2 + s = 0

Routh Method to Find The Number of Instable Root

s 4	4	2	0
s 3	3	1	0
s 2	.6666667	0	
s 1	1	0	
s 0	0		

The Number of The Instable Root is 0

去除一个 $s=0$ 的特征根，得 $4s^3+3s^2+2s+1=0$，经 $d=-0.1$ 平移，得 $4s^3+1.8s^2+1.52s+0.826=0$。

用劳斯判据进行判别，结果如下：

4s^3 + 1.8s^2 + 1.52s + .826 = 0

Routh Method to Find The Number of Instable Root

s 3	4	1.52
s 2	1.8	.826
s 1	－.3155556	
s 0	.826	

The Number of The Instable Root is 2

这表明，原系统的闭环极点除了 $s=0$ 的点外，在 $s=0$ 到 $s=-0.1$ 间还有两个极点。

〔例 5－6〕 开环系统是 $G(s)=\dfrac{K}{(s+1)(s+2)(s+3)}$，采用单位反馈。则得闭环

特征方程是：

$$K + (s+1)(s+2)(s+3) = 0$$

若 $K=15$，得闭环特征方程是：

$$s^3 + 6s^2 + 11s + 21 = 0$$

劳斯判据仿真结果表明该系统是稳定的。

s^3 + 6s^2 + 11s + 21 = 0

Routh Method to Find The Number of Instable Root

s 3	1	11
s 2	6	21
s 1	7.5	
s 0	21	

The Number of The Instable Root is 0

经虚轴平移，可以发现，当 $d=-0.7$ 时系统有不稳定的两个根。而 $d=-0.699$ 时，系统还是稳定的。下面是 $d=-0.699$ 与 $d=-0.7$ 时的仿真结果。

s^3 + 3.903s^2 + 4.077803s + 15.90107 = 0

Routh Method to Find The Number of Instable Root

s 3	1	4.077803
s 2	3.903	15.90107
s 1	3.739928E - 03	
s 0	15.90107	

The Number of The Instable Root is 0

s^3 + 3.9s^2 + 4.07s + 15.897 = 0

Routh Method to Find The Number of Instable Root

s 3	1	4.07
s 2	3.9	15.897
s 1	-6.153645E - 03	
s 0	15.897	

The Number of The Instable Root is 2

这表明，闭环系统的极点有两个在 $d=-0.699$ 到 $d=-0.7$ 间。如在一定的精确度范围内进行平移及判别，就可以确定闭环特征根的负实部的数值。如本例中，有一对共轭复根，其实部为 -0.7。由此可以得到近似的过渡过程时间为 $\frac{3}{\alpha}=4.3$（min）。式中 $\alpha=0.7$。

由此可以看到，若系统的放大倍数 K 先由衰减频率特性来确定，然后用虚轴平移和劳斯判据确定根的负实部，即 $-\alpha$，则根据所给的衰减度即可求出振荡频率等。

应该指出，劳斯判据是代数判据，它不适用于含纯滞后系统的稳定性判别。

5.4 多变量频域的逆奈氏阵列法

由卢森布劳克（Rosenbrock），麦克法伦（MacFarlane）和欧文斯（Owens）等学者创建的现代频域法设计多变量控制系统的方法是把多变量系统的关联减小，从而转化为单变量控制系统的设计方法，由于它们保留了古典频域法的简单、直观的优点，又能借助于计算机进行辅助设计，从而为多变量控制系统的分析和设计提供了一条有效的途径。本节主要介绍卢森布劳克的逆奈氏阵列法。

5.4.1 基本概念

5.4.1.1 矩阵的对角优势

逆奈氏阵列法（INA）的基本思想是在原系统的传递函数矩阵上加入一个补偿器矩阵，使合成为对角优势矩阵，从而实现多变量系统的解耦，然后再按单变量系统进行设计。

对于复数域上的 $n\times n$ 维矩阵 Q，如果满足：

$$\sum_{\substack{j=1\\ j\neq i}}^{m} |q_{ij}| < |q_{ii}| \quad i=1,\cdots,n \tag{5-28}$$

则称 Q 是行对角优势矩阵，若满足：

$$\sum_{\substack{j=1\\ j\neq i}}^{n} |q_{ji}| < |q_{ii}| \quad i=1,\cdots,n \tag{5-29}$$

则称 Q 是列对角优势矩阵。

对于对角优势矩阵，有下列性质满足。

①Q 是行（或列）对角优势矩阵，则 Q 必为非奇异矩阵。其逆 Q^{-1}必存在。

②Q 是行对角优势矩阵，则有

$$\sum_{\substack{k=1\\ k\neq i}}^{n} |q_{ik}| = Q_i |q_{ii}| \quad i=1,\cdots,n \tag{5-30}$$

其中，$0\leqslant\theta_i<1$。其逆矩阵 $q_{ij}^{-1}=q_{ij}$也满足：

$$|q_{ji}|\leqslant\theta_j|q_{ii}| \quad i=1,\cdots,n;j=1,\cdots,n(j\neq i)。 \tag{5-31}$$

若 Q 是列对角优势矩阵，同样有

$$\sum_{k=1}^{n} |q_{ki}| = \theta_i' |q_{ii}| \quad i=1,\cdots,n \tag{5-32}$$

其中，$0\leqslant\theta_i'<1$。逆矩阵 $\hat{Q}=Q^{-1}$的元素 q_{ij}满足：

$$|q_{ij}|\leqslant\theta_j'|q_{ii}| \quad i=1,\cdots,n;j=1,2,\cdots,n(j\neq i)。 \tag{5-33}$$

③奥氏定理（Ostrowski）：

Q 是行对角优势矩阵，其逆为 $\hat{Q}$，则有：

$$|q_{ii}-\hat{q}_{ii}^{-1}|\leqslant\phi_i d_i<d_i \quad i=1,\cdots,n \tag{5-34}$$

其中，$d_i=\sum_{\substack{j=1\\ j\neq i}}^{n}|q_{ij}|$；$\phi_i=\max_{j\neq i}\frac{d_j}{|q_{jj}|}$ (5-35)

若 Q 是列对角优势矩阵，其逆为 $\hat{Q}$，则有：

$$|q_{ii}-\hat{q}_{ii}^{-1}|\leqslant\phi_i' d_i'<d_i' \quad i=1,\cdots,n \tag{5-36}$$

其中，$d_i' = \sum_{\substack{j=1 \\ j\neq i}}^{n} |q_{ji}|$；$\phi_i' = \max_{j\neq i} \frac{d_j'}{|q_{jj}|}$ (5-37)

5.4.1.2 格希哥仁定理（Gershgorin）

复数域内的 $n \times n$ 维矩阵 Q，其特征值在复平面中，以 q_{ii} 为圆心，以 $\sum_{\substack{j=1 \\ j\neq i}}^{n} |q_{ij}|$ 为半径的几个圆的并集中。由于特征值不随矩阵的转置而改变，因此，这些特征值也在以 q_{ii} 为圆心，以 $\sum_{\substack{j=1 \\ j\neq i}}^{n} |q_{ji}|$ 为半径的 n 个圆的并集中。这就是格希哥仁定理。称第一个圆集为格希哥仁行集，第二个圆集为格希哥仁列集。

以 s 平面上的 $-jR$ 到 $+jR$ 的虚轴和右半平面内以 R 为半径的半圆组成一条围线 D，当 R 足够大，使得特征多项式 $\phi(s)$ 的所有右半平面零点均在 D 内，而虚轴上的零点则用 D 在零点附近凸进左半平面进行处理。这样的围线 D 称为奈奎斯特围线。当 s 沿 D 顺时针取值一圈时，围线 D 在特征多项式 $\phi(s)$ 的作用下映射到 $\phi(s)$ 平面的像 Γ_ϕ 称为 $\phi(s)$ 的奈奎斯特图。

若 $Q(s)$ 是定义在 s 的实有理函数域，当 s 在复平面上沿围线 D 顺时针移动一周时，格希哥仁圆的圆心和半径将随 s 的变化而变化，它们将在复平面上组成一个点集。称在这些格希哥仁圆内和圆上的全部点集为格希哥仁带。

在 s 平面上的一条围线 D 上，若 $Q(s)$ 是在 s 的实有理函数域取值的 $n \times n$ 维矩阵，则 $Q(s)$ 是对角优势矩阵是指在 D 上的每个 s 值，$Q(s)$ 或者是行对角优势或者是列对角优势。

因此，若 $Q(s)$ 在围线 D 上是对角优势的，则 $K(s)Q(s)$ 也必然是对角优势的，这里，$K(s)$ 是 s 的有理函数，并且 $K(s)$ 在 D 上没有零点和极点。

$Q(s)$ 在 D 上是对角优势的，其物理意义是相应的格希哥仁圆和格希哥仁带将不包含原点。这是因为对于围线 D 上的任意 s 值，$|q_{ii}(s)|$ 表示格希哥仁圆的圆心与原点的距离，根据对角优势的定义，可以知道，这个距离是大于格希哥仁圆的半径的，因此原点不可能在圆上或圆内，也不可能在相应的格希哥仁带中。

〔例 5-7〕 已知对象的传递函数矩阵 $G(s)$ 的逆矩阵 $\hat{G}(s) = G^{-1}(s)$：

$$\hat{G}(s) = \begin{bmatrix} 2s^3 + 9s^2 + 6s + 2 & 2s + 0.25 \\ 1.5s + 0.5 & 4s^4 + 10s^3 + 14s^2 + 10s + 4 \end{bmatrix}$$

作出 $\hat{g}_{11}(s)$、$\hat{g}_{22}(s)$ 的格希哥仁带及奈奎斯特图。

由于围线 D 是关于 s 的实轴对称的。则 $q_{ii}(s)$ 把围线 D 映射到复平面，也就成为实轴对称的奈奎斯特图，相应的格希哥仁带也将是实轴对称的。因此，在绘制格希哥仁带时，只要取 $s = jR$ 的部分围线即可（R 取 0 到 ∞）。

图 5-6 是 $\hat{g}_{11}(s)$ 与 $\hat{g}_{22}(s)$ 的奈奎斯特图及相应的格希哥仁带。

可以看到，当 $s=0$ 时，$\hat{g}_{11}(s)=2$；$\hat{g}_{12}(s)=0.25$，$\hat{g}_{21}(s)=0.5$，$\hat{g}_{22}(s)=4$。因此相应的格希哥仁圆在圆心（2，0），半径 0.25 及圆心（4，0），半径 0.5。表 5-1 是 $\hat{g}_{11}(s)$ 和 $\hat{g}_{22}(s)$ 的格氏圆的圆心和半径数据表。格氏圆的圆心组成的轨线即是相应的奈奎斯特图。格氏圆的并集组成了相应的格希哥仁带。可以看到，这两条格希哥仁带均不包含原点。因此，$\hat{G}(s)$ 是对角优势矩阵。

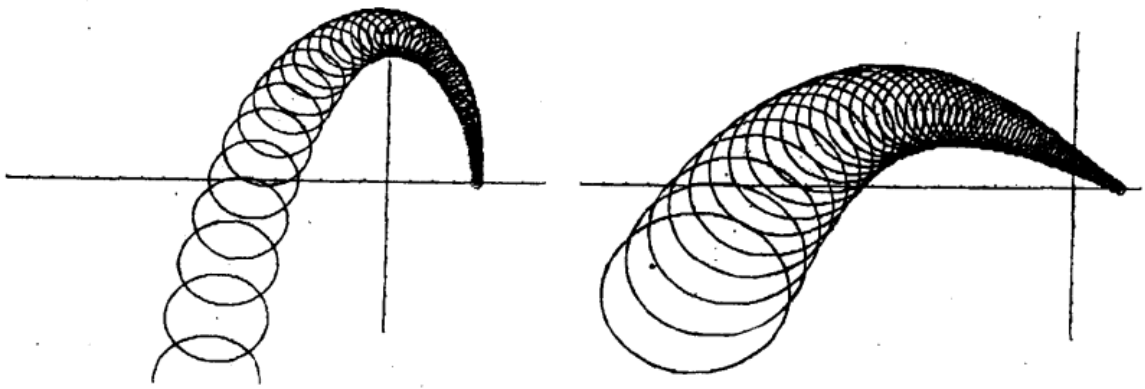


图 5－6　$\hat{G}(s)$ 的格希哥仁带

表 5－1　格希哥仁圆的有关数据

	Inv. of g11(s)		Inv. of g22(s)	
w	Centre	Radius	Centre	Radius
0	2+j 0	.25	4+j 0	.5
.1	1.099+j .598	.32	3.86+j .99	.522
.1258	.867+j .75	.354	3.779+j 1.238	.534
.1584	.574+j .942	.403	3.651+j 1.544	.553
.1995	.204+j 1.181	.47	3.449+j 1.915	.582
.2511	−.26+j 1.474	.56	3.133+j 2.352	.625
.3162	−.846+j 1.833	.68	2.64+j 2.845	.689
.3981	−1.583+j 2.262	.834	1.881+j 3.35	.778
.5011	−2.51+j 2.754	1.032	.736+j 3.752	.902
.6309	−3.679+j 3.283	1.286	−.939+j 3.797	1.07
.7943	−5.149+j 3.763	1.608	−3.241+j 2.931	1.292
1	−7+j 4	2.015	−6+j 0	1.581
1.2589	−9.331+j 3.563	2.53	−8.141−j 7.363	1.953
1.5848	−12.26+j 1.548	3.179	−5.93−j 23.95	2.429
1.9952	−15.95−j 3.914	3.998	11.65−j 59.47	3.034

5.4.2　多变量闭环系统的稳定性

5.4.2.1　多变量闭环系统的结构

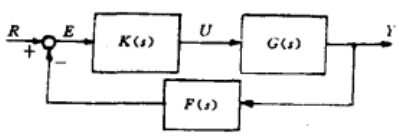


图 5－7　多变量闭环系统的结构

多变量闭环系统的一般结构如图 5－7 所示。图中，R 是 m 维参考输入向量。E 是 m 维的误差向量，Y 是 m 维的输出向量。$G(s)$ 是被控对象的传递函数矩阵，它是 $m \times m$ 维的。$K(s)$ 是 $m \times m$ 维的补偿器传递函数矩阵。$F(s)$ 是 $m \times m$ 维的反馈传递函数矩阵。根据图 5－7，可以得到：

$$Y(s) = G(s)U(s) = G(s)K(s)E(s) = Q(s)E(s) \tag{5-38}$$

式中

$$Q(s) = G(s)K(s) \tag{5-39}$$

称为开环传递函数矩阵。

$\because E(s) = R(s) - F(s)Y(s)$

$\therefore Y(s) = Q(s)E(s) = Q(s)R(s) - Q(s)F(s)Y(s)$

$$\therefore Y(s)=[I+Q(s)F(s)]^{-1}Q(s)R(s)$$

因此，闭环传递函数矩阵 $H(s)$ 是：

$$H(s)=Y(s)R^{-1}(s)=[I+Q(s)F(s)]^{-1}Q(s) \tag{5-40}$$

根据矩阵的运算规则，$H(s)$ 也可以表示为：

$$\begin{aligned}H(s)&=Q(s)Q^{-1}(s)[I+Q(s)F(s)]^{-1}[Q^{-1}(s)]^{-1}\\&=Q(s)\{Q^{-1}(s)[I+Q(s)F(s)]Q(s)\}^{-1}\\&=Q(s)[Q^{-1}(s)Q(s)+Q^{-1}(s)Q(s)F(s)Q(s)]^{-1}\\&=Q(s)[I+F(s)Q(s)]^{-1}\end{aligned} \tag{5-41}$$

进一步转化，$H(s)$ 可表示为：

$$\begin{aligned}H(s)&=[Q^{-1}(s)]^{-1}[I+F(s)Q(s)]^{-1}\\&=\{[I+F(s)Q(s)]Q^{-1}(s)\}^{-1}\\&=[Q^{-1}(s)+F(s)]^{-1}\end{aligned} \tag{5-42}$$

用 $\hat{H}(s)=H^{-1}(s)$，$\hat{Q}(s)=Q^{-1}(s)$ 表示，则（5-42）可写成：

$$\hat{H}(s)=\hat{Q}(s)+F(s) \tag{5-43}$$

通常 $F(s)$ 可以是一个常数阵 F，对于单变量控制系统，通常是单位反馈，即 $F=1$。如果 $F(s)$ 不是常数，可把它的动态部分与 $Q(s)$ 一起考虑。

5.4.2.2 返回差矩阵和特性

如图 5-8 所示，若在闭环回路中断开一点，在 b 点加入输入信号 $\theta(s)$，则在 a 点得到返回信号，输入信号与返回信号的差是

$$\theta(s)-[-Q(s)F(s)\theta(s)]=[I+Q(s)F(s)]\theta(s)=D(s)\theta(s) \tag{5-44}$$

称 $D(s)$ 为返回差矩阵，$D(s)=I+Q(s)F(s)$。 (5-45)

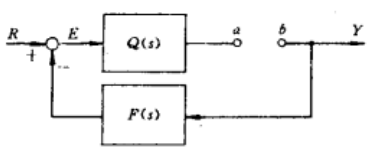


图 5-8 返回差矩阵的结构

称 $T(s)=Q(s)F(s)$ 是返回比矩阵。因此有

$$D(s)=I+T(s) \tag{5-46}$$

对于不同的断点位置，返回差矩阵和返回比矩阵将发生变化，但是，可以证明，返回差矩阵的行列式 $\det[D(s)]$ 的值将不随断点位置改变而改变。

对于单变量系统，如图 5-9 所示。其中，$A(s)$ 与 $B(s)$ 都是 s 的多项式。则系统的开环特征多项式是：

$$\Phi_0(s)=B(s)$$

系统闭环特征多项式可由闭环传递函数

$$H(s)=\frac{Q(s)}{1+Q(s)}=\frac{A(s)}{A(s)+B(s)}$$

求得，即有：

$$\Phi_c(s)=A(s)+B(s)$$

根据返回差矩阵的定义，对于本系统，有：

$$D(s)=1+Q(s)=1+\frac{A(s)}{B(s)}=\frac{A(s)+B(s)}{B(s)}$$

因此，对于单变量系统有

$$D(s)=\frac{A(s)+B(s)}{B(s)}=\frac{\Phi_c(s)}{\Phi_0(s)} \tag{5-47}$$

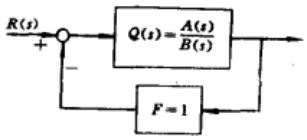


图 5－9　单变量控制系统

对于多变量系统，可以证明，有下列式子成立：

$$| D(s) | = \det[D(s)] = \frac{\Phi_c(s)}{\Phi_0(s)} \qquad (5-48)$$

式中，$|D(s)|$表示返回差矩阵的行列式。$\Phi_c(s)$、$\Phi_0(s)$分别是多变量系统的闭环和开环特征多项式。上式也可以表示为：

$$| D(s) | = \frac{| \hat{H}(s) |}{| \hat{Q}(s) |} = | I + Q(s)F(s) | = | Q(s) | | \hat{Q}(s) + F(s) | \qquad (5-49)$$

5.4.2.3　稳定性判据

多变量系统的稳定性判据可以有多种描述的形式，最常用的是下面的逆奈奎斯特判据。

假设 $F(s) = F = \mathrm{diag}(f_i)$ 是对角矩阵，其元素 f_i 与 s 无关。若开环系统 $Q(s)$ 在右半闭平面有 p_0 个极点，其逆矩阵 $\hat{Q}(s)$ 的对角元素 $q_{ii}(s)$ 的每条格希哥仁带都不包含原点和 $(-f_i, 0)$ 点，它们顺时针围绕原点 $\hat{N}_{qi}$ 次，围绕 $(-f_i, 0)$ 点 $\hat{N}_{hi}$ 次。则闭环系统 $H(s)$ 渐近稳定的充分必要条件是：

$$\sum_{i=1}^{n} \hat{N}_{qi} - \sum_{i=1}^{n} \hat{N}_{hi} = p_0 \qquad (5-50)$$

上述的稳定性判据表明 $\hat{Q}(s)$ 的每条格希哥仁带均不包含原点，也不包含 $(-f_i, 0)$。根据对角优势矩阵的有关知识，可以知道，$\hat{Q}(s)$ 与 $\hat{H}(s)$ 都是满足对角优势条件的。

实际应用时，只需绘制系统开环时传递函数 $Q(s)$ 的逆矩阵 $\hat{Q}(s)$ 的格希哥仁带，就可以判别 $\hat{Q}(s)$ 和 $\hat{H}(s)$ 的对角优势性，同时，根据围绕的次数，可以判别闭环系统的稳定性。

从图 5－6 可以清楚地看到，$g_{11}(s)$ 与 $g_{22}(s)$ 的格希哥仁带不包含原点，因此 $\hat{G}(s)$ 是对角优势的。如果设 $F = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix}$，并满足 $0 \leqslant f_1 \leqslant 9.2$ 及 $0 \leqslant f_2 \leqslant 4$，则可以看到闭环系统 $\hat{H}(s)$ 也是对角优势的。并且由于开环系统的 $p_0 = 0$，因此，这时闭环系统也是稳定的。

逆奈奎斯特判据说明，若开环系统是稳定的（即 $p_0 = 0$），且 $\hat{Q}(s)$ 是在 D 上的对角优势阵，即 $\hat{Q}(s)$ 的各条格希哥仁带不包含原点，那么，这些带的内包络线与 s 平面负实轴的交点到原点的区间就是系统的各个回路增益 f_i 可以取值的区间。因为，这时，可以满足不包含 $(-f_i, 0)$ 的条件，从而使 $\hat{H}(s) = F(s) + \hat{Q}(s)$ 在 D 上是对角优势阵。进而，由于围绕的次数 $\hat{N}_{qi}$ 与 $\hat{N}_{hi}$ 相同，因此，闭环系统的稳定性条件满足。可以看到，稳定性判据的前提条件是 $\hat{Q}(s)$ 与 $\hat{H}(s)$ 在 D 上是对角优势阵。这是它的一个局限性，可通过应用奥氏定理部分弥补这个缺点。

5.4.3　补偿器的设计

由于逆奈奎斯特稳定判据是以 $\hat{Q}(s)$ 和 $\hat{H}(s)$ 是对角优势阵为前提的。因此，多变量系统补偿器设计的首要任务是使系统的传递函数矩阵成为对角优势矩阵，使各系统之间的关联减到允许的程度，然后，对每个回路按单变量系统的设计方法进行设计。本节主要讨论对角优势的实现问题。

5.4.3.1 补偿器的结构

根据卢森布劳克的方法，补偿器 $K(s)$ 的结构应由三部分组成。他指出，若 $\det K(s)\neq 0$，$K(s)$ 的全部零、极点都在 s 的左开半平面，则有：

$$K(s) = K_a \cdot K_b(s) \cdot K_c(s) \tag{5-51}$$

其中三个因子的作用如下述：

K_a：置换矩阵。矩阵的每行和每列中，除了有一个元素是 1 外，其余全为 0。其行列式值等于 ± 1。它的逆矩阵是其自身的转置矩阵，因此，也是一个置换矩阵。当用它左乘另一矩阵时，相当于置换了那个矩阵的行的次序。当用它右乘另一矩阵时，相当于置换了那个矩阵的列的次序。例如，若

$$K_a = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

则 $G(s)K_a$ 的作用相当于把 $G(s)$ 的第一列与第二列相交换，或者相当于把 $G(s)$ 输入的二个编号互相交换。

K_b：初等变换矩阵。行列式等于 1。当用它左乘另一矩阵时，表示进行某种基本的行运算，即把某行乘以某数加到另一行去。例如把等 j 行的 $m(s)$ 倍加到第 i 行时，$K_b(s)$ 就是一个在 (i,j) 位置有元素 $m(s)$ 的单位矩阵。$m(s)$ 的全部极点均在 s 的左开半平面。这种基本矩阵的逆矩阵除了 $m(s)$ 的符号相反外，其余与原矩阵相同。

K_a、$K_b(s)$ 的作用是使 $G(s)K_aK_b(s)$ 成为对角优势矩阵。

$K_c(s)$：非奇异的对角矩阵。每个对角元的零点和极点都在左开半平面。如记 $K_c(s) = \operatorname{diag} k_i(s)$，则

$$K_c(s) = \operatorname{diag} k_i^{-1}(s) = \operatorname{diag} \hat{k}_i(s)$$

即其逆具有相同的特性。

它的作用是对 n 个单回路进行补偿，其设计与单变量系统的设计相同。

$\because Q(s) = G(s)K(s) = G(s)K_aK_b(s)K_c(s)$

$\therefore \hat{Q}(s) = \hat{K}_c(s)\hat{K}_b(s)\hat{K}_a\hat{G}(s)$

如上所述，$\hat{K}_a$ 左乘另一矩阵，相当于对该矩阵进行行变换。$\hat{K}_b(s)$ 左乘一个矩阵，相当于进行行的运算。$\hat{K}_b(s)\hat{K}_a\hat{G}(s)$ 的作用是使其成为对角优势矩阵。

5.4.3.2 补偿器设计实例

下面以两个实例说明补偿器的设计方法。

〔例 5-8〕 研究 $G(s)$ 的对角优势性。已知 $G(s)$ 如下：

$$G(s) = \begin{bmatrix} \dfrac{1}{s+1} & 1 \\ -1 & \dfrac{1}{0.5s+1} \end{bmatrix}$$

提取公分母 $(s+1)(0.5s+1)$ 后，$G(s)$ 改写为：

$$G(s) = \frac{1}{(s+1)(0.5s+1)} \begin{bmatrix} 0.5s+1 & (s+1)(0.5s+1) \\ -(s+1)(0.5s+1) & (s+1) \end{bmatrix}$$

因此，求得 $\hat{G}(s)$ 为：

$$\hat{G}(s) = \frac{1}{0.5s^2+1.5s+2} \begin{bmatrix} (s+1) & -(0.5s+1)(s+1) \\ (s+1)(0.5s+1) & (0.5s+1) \end{bmatrix}$$

从图 5－10 所示的奈氏图和格希哥仁带可以看到系统不是对角优势的。但从图上可以看到如果进行行变换，则可能满足要求。为此，可取 $K_a=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$，相应的奈氏图和格希哥仁带示于图 5－11。系统的稳定性明显改善。

〔例 5－9〕　已知被控对象$G(s)$是

$$G(s)=\begin{bmatrix}\dfrac{1}{s+1} & \dfrac{0.75}{2s+1}\\ \dfrac{0.75}{2s+1} & \dfrac{1}{0.5s+1}\end{bmatrix}$$

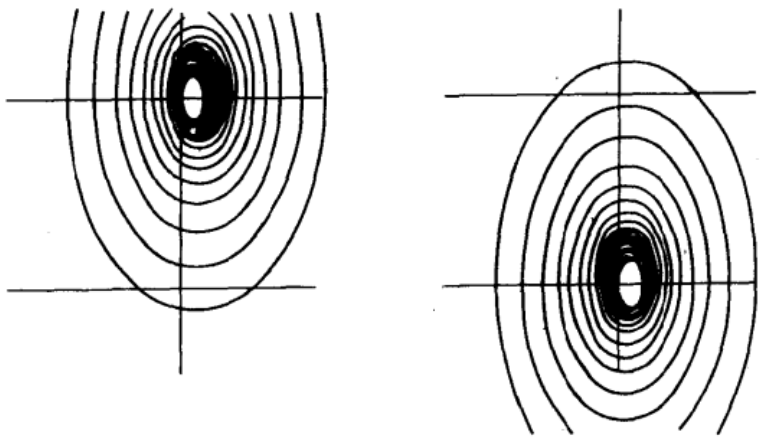


图 5－10　例 5－8 的$G(s)$及格希哥仁带

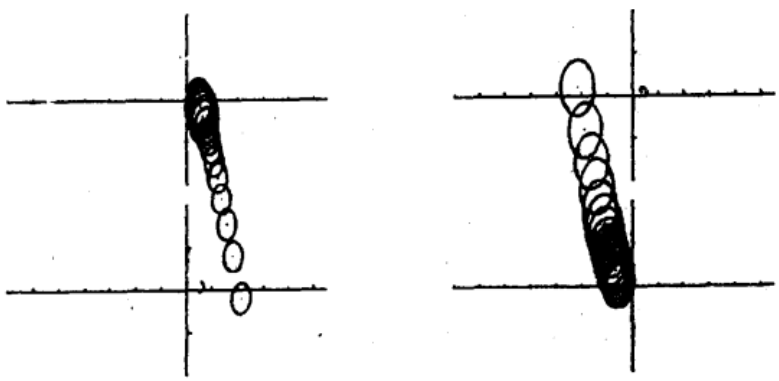


图 5－11　例 5－8 改进后的情况

$G(s)$的公分母是$(s+1)(2s+1)(0.5s+1)$，$G(s)$改写为：

$$G(s)=\frac{1}{(s+1)(2s+1)(0.5s+1)}\begin{bmatrix}(0.5s+1)(2s+1) & 0.75(s+1)(0.5s+1)\\ 0.75(s+1)(0.5s+1) & (s+1)(2s+1)\end{bmatrix}$$

逆矩阵$\hat{G}(s)$为：

$$\hat{G}(s)=\frac{2.29(2s+1)}{(5.7312s+1)(1.4832s+1)}\times\begin{bmatrix}(s+1)(2s+1) & -0.75(s+1)(0.5s+1)\\ -0.75(s+1)(0.5s+1) & (0.5s+1)(2s+1)\end{bmatrix}$$

相应的奈氏图和格希哥仁带示于图 5－12。可以看到，$\hat{G}(s)$的格希哥仁带不包含原点，是对角优势矩阵。但为了改善其特性，也可进行补偿。其 K_a 可取单位矩阵，$K_b(s)$取 $\hat{G}(s)$在原点 $s=0$ 的逆，使在原点邻域总能保证是对角优势。为了使行列式 $\det K(s)=1$，$K_b(s)$取值为：

$$K_b(s) = G(0)/\sqrt{\det G(0)}$$

本例中，$K_b(s)K_a = \begin{pmatrix} 1.51 & 1.13 \\ 1.13 & 1.51 \end{pmatrix}$。

$$\therefore K_b(s)K_a\hat{G}(s) = \frac{2.29}{(5.7312s+1)}\frac{(2s+1)}{(1.4832s+1)} \times \begin{pmatrix} (s+1)(2.595s+0.66) & (0.5s+1)(1.1275s) \\ 1.69875s(s+1) & (0.5s+1)(2.17s+0.66) \end{pmatrix}$$

按上述方式进行补偿后，得到图 5－13（a）所示的格希哥仁带。其特性反而变差。

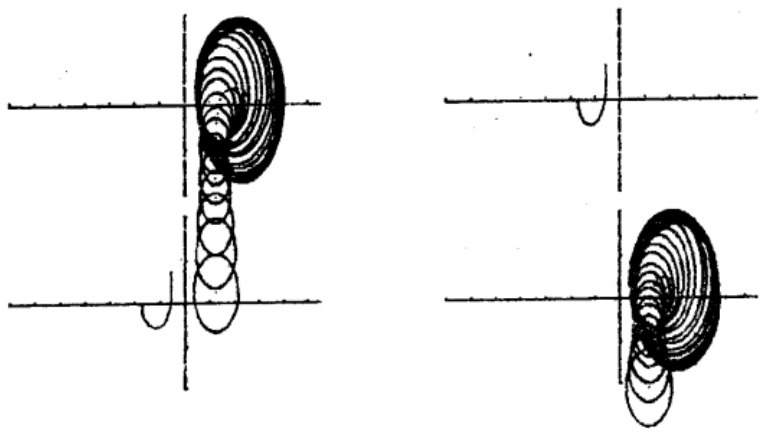


图 5－12　例 5－9 的格希哥仁带

一个解决的方法是用 $s=j\omega$ 代入，$K_b(s)$成为：

$$K_b(s) = |G(s)|_{s=j\omega} \cdot [\det|G(s)|_{s=j\omega}]^{1/2}$$

频率 ω 采用试凑的方法。例如，本例中，取 $\omega=1$ 及 $\omega=2$ 得到图 5－13（b）及（c）的格希哥氏带。

比较后可以得到，当用 $s=j\cdot 1$ 时有较好的特性，由此，求得

$$G(s)K_aK_b(s) = \begin{pmatrix} \frac{1}{s+1} & \frac{0.75}{2s+1} \\ \frac{0.75}{2s+1} & \frac{1}{0.5s+1} \end{pmatrix} \begin{pmatrix} 1.24 & -0.46 \\ -0.46 & 0.98 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{0.895(2.385s+1)}{(s+1)(2s+1)} & -\frac{0.275(0.673s-1)}{(s+1)(2s+1)} \\ -\frac{0.47(0.968s-1)}{(2s+1)(0.5s+1)} & \frac{0.635(2.815s+1)}{(2s+1)(0.5s+1)} \end{pmatrix}$$

经补偿后的系统已为对角优势系统，特性已经改善，余下的问题是选择对角矩阵 $K_c(s)$，这只要根据单输入单输出控制系统的要求进行设计即可。

因此，用逆奈奎斯特阵列法设计多变量控制系统的问题就是选择合适的$K(s)$，使 $\hat{Q}(s)=K(s)\hat{G}(s)$成为对角优势的。然后根据 $H(s)=\hat{Q}(s)+F$，可以确定 $H(s)$的对角优势，并

确定反馈阵的增益。对于 $\hat{K}(s)$的设计有多种方法。通常可采用下列方法：

①选择$\hat{K}(s)$是常数矩阵 K。当$\hat{G}(s)$本身已是 对角优势阵时，可选$\hat{K}(s)=I$。一种常用的方法是选$\hat{K}_b(s)\hat{K}_a=|G(s)|_{s=j\omega}\cdot[\det|G(s)|_{s=j\omega}]^{1/2}$，并对不同的 ω 进行试凑。

②选择$\hat{K}(s)$是超前或滞后网络矩阵。采用试凑的方法确定 $\hat{K}(s)$，它对高、低频都有影响。

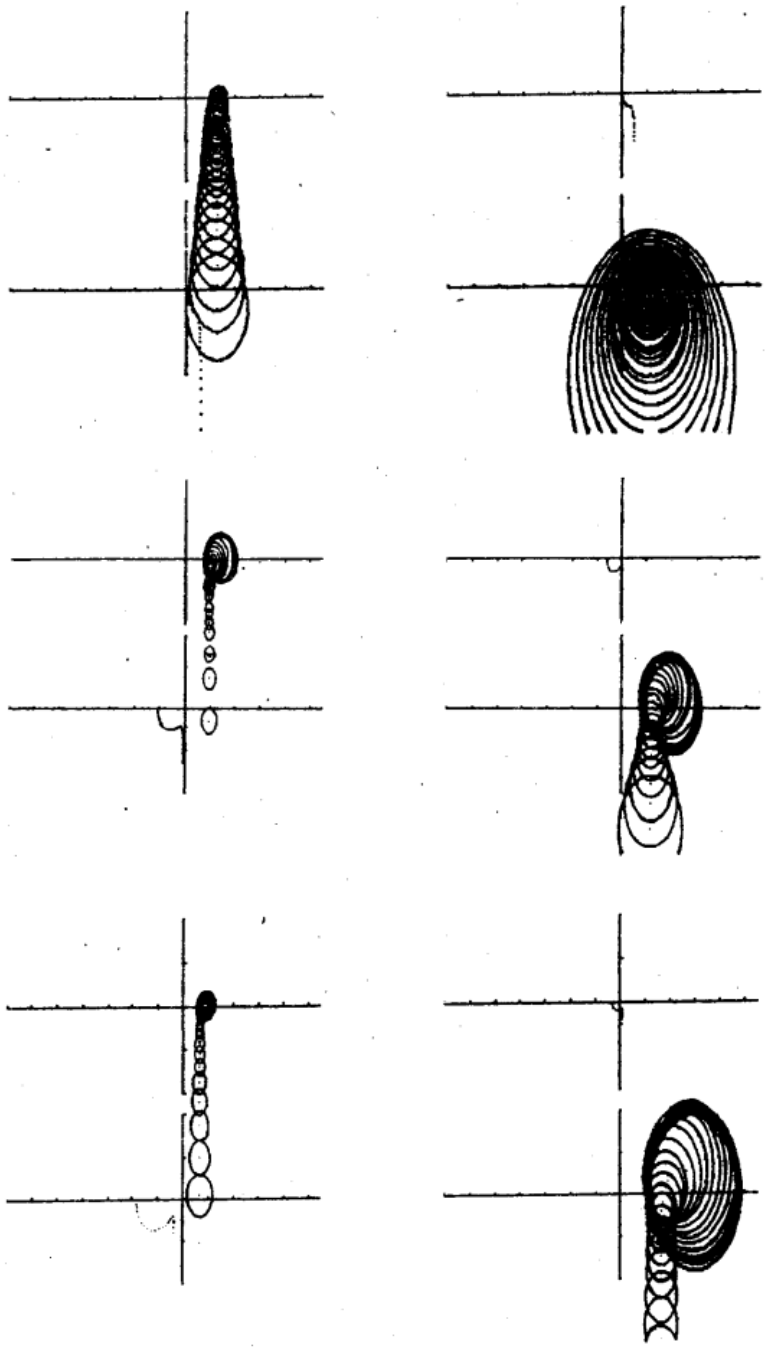


图 5－13　例 5－9　的改进情况

③选择$\hat{K}(s)$是两个矩阵之和。其中一个是在低频时起对角优势作用的滞后网络，另一个是在高频时起对角优势作用的超前网络。

应该指出，如果不对$\hat{K}(s)$加以限制，最简单的方法是使$\hat{K}(s)=G(s)$，则$\hat{Q}(s)=\hat{K}(s)\hat{G}(s)=I$，成为对角优势矩阵。然后，这样设计所得的$K(s)$往往很复杂，而且有时还可能无法在物理上实施。因此，在$\hat{K}(s)$的选择时常常需要有某些限制条件，在设计时应该予以注意。

5.4.3.3 设计步骤和主要算法

根据上面的实例分析，对多变量控制系统的设计步骤说明如下：

①根据$G(s)$，计算$\hat{G}(s)$，并$\hat{K}_1 \leftarrow I$。

②计算$\hat{Q}(s)=\hat{K}_1\hat{G}(s)$。

③绘制$\hat{Q}(s)$的奈奎斯特图及格希哥仁带，如$\hat{Q}(s)$是对角优势阵，则转⑤，否则转④。

④选取一定特定频率ω，首先取$\omega=0$，然后是选定的频率，计算$\hat{K}_1$，并转②。

⑤根据格希哥仁带的包络线与负实轴的交点，选取各回路增益。

⑥计算$K_1=\hat{K}_1^{-1}$，并把K_1送$K(s)$。

⑦计算$Q(s)=G(s)K(s)Q(s)$，或计算$\hat{H}(s)=\hat{Q}(s)+F$。

⑧计算$H(s)=[I+Q(s)F]^{-1}Q(s)$，或用$H(s)=\hat{H}(s)^{-1}$计算。

⑨计算$H(s)$的$n\times n$个传递函数的阶跃响应。

⑩判断闭环系统性能，如满足性能要求，则转⑫。

⑪按单输入单输出的系统设计方法，设计各回路的补偿器$K_c(s)=\mathrm{diag}[K_{c1}(s),K_{c2}(s),\cdots,K_{cn}(s)]$。并计算$K(s)=K_1\cdot K_c(s)$。转⑦。

⑫输出补偿器矩阵$K(s)$。

上述的设计步骤中，如果由计算机来完成全部工作将会使程序变得十分复杂。而且设计效果未必满意。通常可以把计算机的计算、绘图能力和设计人员的综合分析、判断能力相结合。例如，对角优势是否满意，闭环性能是否好，如何选择频率ω等可由设计者完成。其他的设计步骤则由计算机完成。

在上述的设计步骤中主要有下述一些算法。

①有理矩阵的运算：包括有理矩阵的求逆、有理矩阵的相乘等运算。

②计算特定频率下的K_1：这里包括复变数的运算、矩阵行列式值计算等。

③格希哥仁带的绘制：计算奈奎斯特图的对角线元素坐标和根据非对角线元素计算半径。由于IBM－PC机有图形语句CIRCLE，因此，绘制圆形十分方便，但由于CRT分辨率的原因，在CRT上显示时成为椭圆，而硬拷贝时也成为椭圆。虽然如此，由于坐标位置不发生变化，因此不影响对角优势的判别及稳定性的界限确定。

④计算阶跃响应曲线：可以采用第二章或第三章的数字仿真程序计算阶跃响应，绘制相应的阶跃响应曲线。

在上述的设计步骤和算法中，没有论及当$\hat{Q}(s)$包含原点或$(-f_i,0)$的情况，当出现这种情况时，可用奥氏定理判断。与格希哥仁带相类似，根据奥氏定理可以画出相应的奥氏带。奥氏带可实现两个功能。第一个功能是用这些带可以确定逆传递函数$h_i^{-1}(s)$，它是第i个回路开路，所有其他回路按特定增益闭路时第i个输入和第i个输出间传递函数的逆。这样，当其余回路增益改变时，$h_i(s)$也发生变化，即第i个奥氏带也取决于其他回路的增益。而格希哥仁带则和各回路增益无关。

第二个功能是可用它来确定各回路的稳定裕度。更详细的内容可参考有关文献。

6 最优控制系统的数字仿真

最优控制系统的数字仿真是优化算法和控制系统数字仿真的结合。按一定的准则，通过控制系统的数字仿真求得目标函数，采用优化算法调整控制器参数，使目标函数趋向极值是本章讨论的参数优化问题。前面几章已经讨论了控制系统的数字仿真，因此，本章主要介绍各种优化算法，但在实例中介绍了优化算法和控制系统数字仿真结合的方法。最优控制系统数字仿真中的结构优化因篇幅所限，故未作讨论，

6.1 概　　述

6.1.1 目标函数和约束条件

最优控制系统是在满足一定约束条件下，使目标函数趋向极值时的控制系统。根据目标函数的不同形式或不同的约束条件，对于同一个控制系统，其最佳参数值会有不同数值，因此，对目标函数的选择和约束条件的确定应予以讨论和分析。

在设计最优控制系统时，要根据生产过程的具体情况和要求，正确选择目标函数，提出合适的约束条件。只有这样，设计出来的控制系统才能真正在实际应用中有效发挥其作用。

6.1.1.1 目标函数

目标函数的选择，既要考虑到能对控制系统的性能作出正确评价，又要考虑到工程上易于实施和数学上容易处理。

目标函数可以表达为系统性能指标的函数。它应该与控制器的被寻优参数有关。例如，可用控制系统动态响应的性能指标，如过渡时间、超调量、最大偏差等来表示。也可以用误差积分型性能指标来表示。目标函数可以是定常函数，也可以是时变函数。常用的目标函数有：

①积分型：按误差准则建立的积分型目标函数有很多种形式。如：

$$ISE = \int_0^\infty e^2(t)dt \tag{6-1}$$

$$ITSE = \int_0^\infty te^2(t)dt \tag{6-2}$$

$$IAE = \int_0^\infty |e(t)|dt \tag{6-3}$$

$$ITAE = \int_0^\infty t|e(t)|dt \tag{6-4}$$

ISE 指标着重于大的误差的消除，对于阶跃信号输入，存在大的起始误差时能迅速减小误差，因此响应迅速，相应地，它的系统相对稳定性较差。*ITSE* 性能指标着重于瞬态响应后期出现的误差，对阶跃响应起始段的大误差考虑较少，有较好的选择性。*IAE* 的选择性不太好，可使系统有适当的阻尼和好的瞬态响应。*ITAE* 与 *ITSE* 相似，对起始段大的误差考虑较少，有较好的选择性，系统动态响应的超调量很小，有足够的阻尼。

当目标函数 J 取不同性能指标时，如设计时，取 $J = ISE$ 为极小值，则所得的系统是平方误差积分值为最小值的最优控制系统。该系统对于其他性能指标而言，不一定是最优控制系统。因此，最优控制系统是针对特定的目标函数而言的。

把误差的概念扩大到状态的偏差，若要求系统达到平衡时，系统的状态为 $x_e = 0$，则状态偏差为 $x - x_e = x$，其积分型目标函数可表达为：

$$J = \int_0^\infty x^T Q x dt \tag{6-5}$$

式中，x 是状态向量。Q 是加权矩阵，是反映目标函数 J 中各分量权重的对角矩阵。

更一般的情况是讨论如下的控制系统：

$$\dot{x}(t) = Ax(t) + Bu(t)$$

要求设计控制规律 u (t)，使给定的目标函数

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \tag{6-6}$$

为最小。这里，控制向量$u(t)$是状态向量 $x(t)$的线性组合。称 (6-6) 式所示的目标函数为线性二次型性能指标。式中，Q是正定或半正定的实对称矩阵，R是正定实对称矩阵。

②终值型：当目标函数是终值时间 t_f 和终值状态x (t_f) 的函数时，即

$$J = S[x(t_f), t_f] \tag{6-7}$$

则称 J 为终值型目标函数。

③综合型：把上述两类型结合起来，用

$$J = \int_0^\infty (x^T Q x + u^T R u) dt + S[x(t_f), t_f] \tag{6-8}$$

来表示综合型的目标函数。

6.1.1.2 约束条件

在工业生产过程的控制中，控制向量 $u(t)$ 总是受一定的物理限制，而不能取任意大的值。如调节阀确定后，流过它的流量就有一定的限制。调节阀的开度也有一定的限制。这些限制称为约束条件。如

$$|u_i(t)| \leqslant u_{i\max} \qquad (i = 1, \cdots, m) \tag{6-9}$$

写成集合的形式是

$$u_i(t) \in U \subset R_m$$

U 是 m 维空间 R_m 的一个子集。

此外，控制系统中各状态变量 $x_i(t)$ 在扰动作用下也允许在一定的数值范围变化，即有

$$|x_i(t)| \leqslant x_{i\max} \qquad (i = 1, \cdots, n) \tag{6-10}$$

或写成

$$x_i(t) \in X \subset R_n$$

约束条件要与目标函数统一考虑，例如，当用误差积分和作为目标函数时，一个具有等幅振荡的系统可能会有很小的目标函数值，这时通过加入误差的约束条件，就能保证一定的控制精度。因此，选择合适的目标函数和约束条件是十分必要的。

6.1.2 最优控制系统的数学描述

最优控制系统的设计可分为两类，不考虑时间因素影响的最优控制系统的设计是静态最

优控制设计。反之，则称为动态最优控制设计。

6.1.2.1 静态最优控制

静态最优控制忽略了对象动态特性对整个系统的影响，这对于大多数工业过程是可以近似适用的，且有足够的精度。因此，本章将介绍适用于静态最优控制设计的线性规划法、爬山法及优选法等。

不考虑时间因素的静态最优控制设计问题可以用一个目标函数 $J = f(x)$ 和若干个约束条件

$$g_i(x) = 0 \qquad i = 1,2\cdots, m$$

$$h_j(x) \leqslant 0 \qquad j = 1,2\cdots, p$$

来描述。要求设计控制系统在满足约束条件下，使目标函数 J 为最小（或最大）。这里，$g_i(x) = 0$ 称等式约束条件。$h_j(x) \leqslant 0$ 称不等式约束条件。目标函数 J 和约束条件中的变量均与时间无关。

当目标函数和约束条件可以用线性数学式描述时，可以用线性规划的方法求取最优解。当它们不能直接求得，但可用测量装置知道它们的大小时，常可用爬山法求取近似的最优解。如生产过程的状态变量及操纵变量允许作大幅度的变化，而且生产过程的运行允许时，常可采用优选法求解。

6.1.2.2 动态最优控制

动态最优控制多数用于时间过程影响明显的过程，例如在分级控制系统中，级数处于下面的系统受时间过程的影响较明显。

动态最优控制问题的数学描述是：在一定的约束条件下，求满足控制系统状态方程

$$\dot{x}_i = f_i[x(t), u(t), t] \qquad (i = 1,2,\cdots, n)$$

及使目标函数

$$J = \int_{t_0}^{t_f} F[x(t), u(t), t]dt$$

为最小的最优操纵变量 $u^*(t)$。这里，目标函数 J 是函数 $x(t)$、$u(t)$ 的函数，因此，称为泛函数，简称泛函。

当操纵变量的约束条件对求解过程没有影响时，可以采用变分法求解。反之，则可用极大值原理求解。当 $F=1$ 时，目标函数成为求最短时间，这时可以用最短时间控制的方法求解。应该指出，极大值原理包括了变分法及最短时间控制的内容。动态规划法弥补了变分法求解时要知道系统初始状态及伴随系统初态的缺陷，它采用最优化原则从后级逐级进行最优决策，从而求取最优解，因此在工业生产过程的调度、负荷分配等最优控制系统的设计中得到了应用。

6.1.3 目标函数取极值的充要条件

最优控制的设计问题其实质是在约束条件下目标函数的求极值问题。因此，下面介绍目标函数取极值的充分必要条件。

设目标函数 J

$$J = f(x)$$

这里，$x = [x_1, x_2, \cdots, x_n]^T$，是 n 维列向量。

J 取极值的必要条件是

$$\frac{\partial f}{\partial x} = 0 \tag{6-11}$$

即函数 $f(x)$ 的梯度 $\mathrm{grad}f(x)$ 是零向量。

$$\mathrm{grad}f(x) = \left[\frac{\partial f}{\partial x_1}\ \frac{\partial f}{\partial x_2}\ \cdots\ \frac{\partial f}{\partial x_n}\right]^T = 0 \tag{6-12}$$

J 有极值的充分条件是汉森（Hession）矩阵

$$\frac{\partial^2 f}{\partial x^2} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & & \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{6-13}$$

是正定矩阵（J 有极小值）或负定矩阵（J 有极大值）。

与此相对应，以泛函 J 表示的目标函数

$$J = J(X)$$

的极值存在的必要条件是 J 的一次变分

$$\delta J(X) = 0$$

而其充分条件是 J 的二次变分

$\delta^2 J(X) \leqslant 0$（极大值）或 $\delta^2 J(X) \geqslant 0$（极小值）

6.2 静态最优化方法

6.2.1 线性规划方法

6.2.1.1 凸集及顶点

若在 n 维空间中的点集 C，满足在集 C 中任意两点的联线属于 C，则称点集 C 是凸集。如果凸集上任意一点，不能成为该集上某一线段的中点，则称为顶点。

凸集的顶点可以看成是线性方程组的一个解，凸集 C 上的点 $x^j = [x_1^j, x_2^j, \cdots, x_n^j]$ 是顶点的充要条件是 x^j 的非零坐标 x_1^j，x_2^j，$\cdots$，x_m^j 所对应的向量 p_1，p_2，$\cdots$，p_m 是与线性无关的，其中 j 表示属于第 j 个顶点。对于已给定的凸集，它的顶点个数是有限的。

线性规划方法是一种最优化方法，它的目标函数和约束条件都是线性函数。满足线性规划约束条件的点组成一个凸集。凸集的每一个点组成可行解。凸集的顶点称为基本可行解。最优解应该在凸集的顶点处。因此，线性规划寻优的方法就是从一个已知的凸集顶点出发，通过一系列替换，得到另一个顶点，在各顶点计算目标函数值，并使之为最小。

6.2.1.2 图解法

对于两个变量的线性规划问题，可用图解法寻优。在直角坐标系中，画出约束条件所表示的直线方程，组成一个凸多边形，这个凸多边形就是可行解的集合。对于给定的目标函数 J，作出 J 的等值直线，在等值线上 J 有相同的值，距离原点越远，等值线 J 的值较大，从而可在凸多边形的某个顶点找到 J 的最大值。

6.2.1.3 单纯形法（Simplex Method）

实际的求解最优问题中，目标函数 J 可能是极大，也可能是极小，这里若以求极小进行讨论，则求极大值的问题只需用 $-J$ 作为目标函数。

单纯形法寻优的过程是从一个初始顶点出发，按一定规则进行顶点的替换，使目标函数值不断减小，直到求得最优点。其计算步骤如下：

①引入松弛变量，使不等式约束条件成为等式约束条件。写成矩阵形式是：

$$Ax = b$$

系数矩阵 A 是 $m \times n$ 矩阵，$n = N + m$，N 为原变量（结构变量）个数，m 是松弛变量个数。

$$A = \left[\begin{array}{ccc|ccc} a_{11} & \cdots & a_{1N} & 1 & & 0 \\ \vdots & & & & 1 & \\ & & & & & \ddots \\ a_{m1} & \cdots & a_{mN} & 0 & & 1 \end{array}\right] \tag{6-14}$$

$$x = [x_1, x_2, \cdots, x_N, x_{N+1}, \cdots, x_{N+m}]^T \tag{6-15}$$

$x_1, \cdots x_N$ 为结构变量，$x_{N+1}, \cdots, x_{N+m}$ 是松弛变量。

②以松弛变量作为替换过程的基底变量，以结构变量作为非基底变量组成初始顶点。

$$x = [x_1, \cdots, x_N, x_{N+1}, \cdots, x_{N+m}] = [0, \cdots, 0, b_1, \cdots, b_m]$$

由于约束条件是

$$\sum_{i=1}^{N} a_{ji} x_i \leqslant b_j \tag{6-16}$$

$$\therefore \quad x_{N+j} = b_j - \sum_{i=1}^{N} a_{ji} x_i \geqslant 0$$

③计算目标函数值：加入松弛变量后的目标函数的系数矩阵成为

$$C = [c_1, c_2, \cdots, c_N, c_{N+1}, \cdots, c_{N+m}] \tag{6-17}$$

其中，$c_{N+1}, \cdots, c_{N+m}$ 均为 0。

$$J_0 = \sum_{i=1}^{N} c_i x_i + \sum_{i=N+1}^{N+m} c_i x_i = \sum_{j=1}^{m} c_{N+j} b_j = 0 \tag{6-18}$$

④替换顶点：用一个非基底变量替换一个基底变量，如用 x_i 替换 x_{N+j}，则替换后的顶点成为：

$$x = [0, \cdots, x_i, \cdots, 0, x_N, \cdots, x_{N+j-1}, 0, x_{N+j+1}, \cdots, x_{N+m}]$$

相应的目标函数 J

$$\begin{aligned} J &= J_0 + \sum_{i=1}^{N} c_i x_i - \sum_{j=1}^{m} c_{N+j} x_{N+j} \\ &= J_0 + \sum_{i=1}^{N} z_i x_i \end{aligned} \tag{6-19}$$

式中，

$$z_i = c_i - \sum_{j=1}^{m} a_{ji} c_{N+j} > 0 \quad (i = 1, \cdots, N) \tag{6-20}$$

为加速求解过程，比较求出最大的 z_i，相应的换入变量记为 x_e。

对换出变量按下列条件选择：

$$\min\left[\frac{b_j}{a_{je}}\right] \quad (\text{对所有 } a_{je} > 0) j = 1, \cdots, m \tag{6-21}$$

记相应的换出变量为 x_{N+r}。则经替换后，结构表的系数要重新计算。计算公式如下：

$$a_{re}^{K+1}=\frac{1}{a_{re}^{K}} \tag{6-22}$$

$$a_{ri}^{K+1}=\frac{a_{ri}^{K}}{a_{re}^{K}} \tag{6-23}$$

$$a_{ji}^{K+1}=a_{ji}^{K}-a_{ri}^{K}\frac{a_{je}^{K}}{a_{re}^{K}}\quad(j=1,\cdots,m;j\neq r) \tag{6-24}$$

$$b_{j}^{K+1}=a_{j0}^{K}-a_{r0}^{K}\frac{a_{je}^{K}}{a_{re}^{K}}\quad(j=1,\cdots,m;j\neq r) \tag{6-25}$$

式中 K 是第 K 次替换后的计算值。

新的 z_i^{K+1} 由 $z_i^{K+1}=c_i^{K+1}-\sum_{j=1}^{m}a_{ji}^{K+1}c_j^{K+1}$ 确定。

⑤判别 z_i，当求 J 的极小值时，z_i 取小于0，反之取 $z_i>0$。满足条件时，则得最优解，否则转④再替换另一个顶点进行计算。

6.2.1.4 线性规划仿真程序

下面给出用单纯形法求解线性规划的仿真程序。程序中变量说明如下：

N：原变量个数。

M：约束条件方程个数。

P：求极值标志。P=1 求极大值。P=－1 求极小值。

PR：显示标志。PR=1 表示数据打印并在 CRT 显示。PR≠1 则表示仅在 CRT 显示。

A（I，J.）：约束条件中各变量前的系数矩阵。

$\begin{bmatrix} a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & & & \vdots \\ a_{M+1,1} & a_{M+1,2} & \cdots & a_{M+1,N} \end{bmatrix}$ 是约束条件的相应变量前的系数。

$[a_{2,N+1}\cdots a_{M+1,N+1}]^T$ 表示约束条件标志。其值为－1，表示约束条件是≥。其值为0表示=。其值为1表示≤。

$[a_{2,N+2}\cdots a_{M+1,N+2}]^T$ 表示约束条件右边的数值。

$[a_{11}\cdots a_{1N}]$ 是目标函数的相应变量前系数。

数据输入用 DATA 语句。格式是

M,N,P,A(2,1),A(2,2),…,A(2,N+2),A(3,1),…A(3,N+2),…,A(M+1,1),…A(M+1,N+2),A(1,1),A(1,2),…A(1,N)。

例如

2000 DATA 3,2,－1,1,0,1,－1,2,0,1,2,－1,5,4,3,8

表示有3个变量，2个约束条件方程的求极小值的线性规划问题。其约束条件是

$$\begin{cases} x(1)+x(3)\geqslant 2 \\ x(2)+2x(3)\geqslant 5 \end{cases}$$

目标函数 $J=4x(1)+3x(2)+8x(3)$

打入 DATA 的该语句后，即可运行仿真程序。在 CRT 上会询问

"Display Option 1:Print & CRT 0:CRT

打入1表示打印并显示。0则仅显示在CRT。

打入 1 前应打开打印机，连好与计算机的连接电缆，并准备好打印纸。打入 1 后，即会打印出如下结果：

```
x(1) + x(3) > = 2
x(2) + 2 * x(3) > = 5
J = 4 * x(1) + 3 * x(2) + 8 * x(3)
.....

Object Function Jmin = 19
x(1) = 0
x(2) = 1
x(3) = 2
```

其中，前面的部分是线性规划的问题(即约束条件和目标函数)后面部分是目标函数的极值及相应的最优解。

```
1000  CLS
1005  REM N:Variable Number;M:Constraints Number
1010  LOCATE 2,1
1015  PRINT "Simplex method to solve LP problem"
1020  LOCATE 4,1
1025  PRINT "Reading Data Now!":REM N,M,P and A(i,j),J
1030  LOCATE 6,1
1035  PRINT "Display Option 1:Print & CRT 0:CRT"
1040  INPUT PR:REM PR:Flag (1 or0)
1045  IF PR<0 OR PR>1 THEN 1030
1050  READ N,M,P:REM P:Flag (1 or -1) 1:Max -1:Min
1055  DIM A(M+2,N+M+1),X(N),L(M+1)
1060  FOR I=2 TO M+1
1065  FOR J=1 TO N+2
1070  READ A(I,J):REM Matrix A
1075  NEXT J
1080  NEXT I
1085  FOR I=2 TO M+1
1090  FOR J=1 TO N+2
1095  A(I,J)=A(I,J) * SGN(A(I,N+2))
1100  NEXT J
1105  NEXT I
1110  FOR I=1 TO N
1115  READ A(1,I):REM Object Function
1120  NEXT I
1125  FOR I=2 TO M+1
1130  B=1
1135  IF PR=1 THEN LPRINT
```

```
1140  FOR J=1 TO N
1145  IF A(I,J)>0 AND J<>B THEN PRINT "+";
1150  IF A(I,J)>0 AND J<>B AND PR=1 THEN LPRINT "+";
1155  IF A(I,J)<0 THEN PRINT "-";
1160  IF A(I,J)<0 AND PR=1 THEN LPRINT "-";
1165  IF A(I,J)=0 THEN B=B+1:GOTO 1195
1170  IF A(I,J)=1 THEN LPRINT " ";:GOTO 1185
1175  PRINT A(I,J);"*";
1180  IF PR=1 THEN LPRINT A(I,J);"*";
1185  IF PR=1 THEN LPRINT "x(";J;")";
1190  RPINT "x(";J;")";
1195  NEXT J
1200   IF A(I,N+1)=-1 THEN PRINT ">=";
1205  IF A(I,N+1)=-1 AND PR=1 THEN LPRINT ">=";
1210  IF A(I,N+1)=0 THEN PRINT "=";
1215  IF A(I,N+1)=0 AND PR=1 THEN LPRINT "=";
1220  IF A(I,N+1)=1 THEN PRINT "<=";
1225  IF A(I,N+1)=1 AND PR=1 THEN LPRINT "<=";
1230  PRINT A(I,N+2)
1235  IF PR=1 THEN LPRINT A(I,N+2)
1240  NEXT I
1245  B=1
1250  IF PR=1 THEN LPRINT
1255  IF PR=1 THEN LPRINT "J=";
1260  PRINT "J=";
1265  FOR I=1 TO N
1270  IF PR=1 AND A(1,I)>0 AND I<>B THEN LPRINT "+";
1275  IF A(1,I)>0 AND I<>B THEN PRINT "+";
1280  IF PR=1 AND A(1,I)<0 THEN LPRINT "-";
1285  IF A(1,I)<0 THEN PRINT "-";
1290  IF PR=1 AND A(1,I)=0 THEN B=B+1:GOTO 1325
1295  IF PR=1 AND A(1,I)=1 THEN 1315
1300  IF A(1,I)=1 THEN 1315
1305  IF PR=1 THEN LPRINT A(1,I);"*";
1310  PRINT A(1,I);"*";
1315  IF PR=1 THEN LPRINT "x(";I;")";
1320  PRINT "x(";I;")";
1325  NEXT I
1330  IF PR=1 THEN LPRINT
1335  PRINT
```

```
1340  FOR I=1 TO N
1345  A(1, I) = -P * A(1, I)
1350  NEXT I
1355  IF M=1 THEN 1395
1360  FOR I=2 TO M+1
1365  A(I, N+M+1) = A(I, N+2)
1370  A(I, N+2) = 0
1375  IF I=2 THEN 1390
1380  A(I, N+I-1) = A(I, N+1)
1385  A(I, N+1) = 0
1390  NEXT I
1395  R=1
1400  FOR I=2 TO M+1
1405  L(I) = N+I-1
1410  IF A(I, N+I-1)=1 THEN 1440
1415  L(I) = N+M+1
1420  FOR J=1 TO N+M
1425  A(M+2, J) = A(M+2, J) - A(I, J)
1430  NEXT J
1435  R=M+2
1440  NEXT I
1445  T=1
1450  FOR I=2 TO N+M
1455  IF A(R, I) - A(R, T) >E THEN 1480
1460  IF A(R, I) - A(R, T) < -E THEN 1475
1465  IF R=1 THEN 1480
1470  IF A(1, I) - A(1, T) >= -E THEN 1480
1475  T=I
1480  NEXT I
1485  IF A(R, T) < -E THEN 1545
1490  IF R=1 THEN 1675
1495  FOR I=1 TO N+M
1500  IF A(R, I) >E THEN 1520
1505  NEXT I
1510  R=1
1515  GOTO 1445
1520  FOR I=2 TO M+1
1525  IF L(I) <=N+M THEN 1535
1530  IF A(I, N+M+1) >E THEN 1655
1535  NEXT I
```

```
1540  GOTO 1675
1545  S=1
1550  FOR I=2 TO M+1
1555  IF A(I, T)<=E THEN 1580
1560  Y=A(I, N+M+1)/A(I, T)
1565  IF S=1 THEN 1575
1570  IF Y>=A(S, N+M+1)/A(S, T) THEN 1580
1575  S=I
1580  NEXT I
1585  IF S=1 THEN 1665
1590  L(S)=T
1595  Y=A(S, T)
1600  FOR I=1 TO N+M+1
1605  A(S, I)=A(S, I)/Y
1610  NEXT I
1615  FOR I=1 TO M+2
1620  IF I=S THEN 1645
1625  Y=A(I, T)
1630  FOR J=1 TO N+M+1
1635  A(I, J)=A(I, J)-Y*A(S, J)
1640  NEXT J
1645  NEXT I
1650  GOTO 1445
1655  PRINT "Infeasible"
1660  GOTO 1765
1665  PRINT "Unbounded"
1670  GOTO 1765
1675  FOR I=2 TO M+1
1680  IF L(I)>N THEN 1695
1685  S=L(I)
1690  X(S)=A(I, N+M+1)
1695  NEXT I
1700  Y=P*A(1, N+M+1)
1705  IF PR=1 THEN LPRINT
1710  IF PR=1 THEN LPRINT"....................................."
1715  IF PR=1 THEN LPRINT
1720  IF PR=1 AND P=-1 THEN LPRINT "Object Function Jmin=";Y
1725  IF PR=1 AND P=1 THEN LPRINT "Object Function Jmax=";Y
1730  B=1
1735  IF PR=1 THEN LPRINT
```

```
1740  PRINT "Object Function J = "; Y
1745  FOR I = 1 TO N
1750  PRINT "x("; I; ") = "; X(I)
1755  IF PR = 1 THEN LPRINT "x("; I; ") = "; X(I)
1760  NEXT I
1765  END
```

〔例 6－1〕 求解下列极大值问题。

$$\begin{cases} 4x(1) + x(2) \leqslant 150 \\ 6x(1) + 5x(2) \leqslant 280 \\ x(1) + 3x(2) \leqslant 90 \end{cases}$$

$$J = 50x(1) + 70x(2)$$

送入数据语句如下：

```
2000  DATA 2, 3, 1, 4, 1, 1, 150, 6, 5, 1, 280, 1, 3, 1, 90, 50, 70
```

运行时，打入 1 后打印结果如下：

```
4 * x(1) + x(2) < = 150
6 * x(1) + 5 * x(2) < = 280
x(1) + 3 * x(2) < = 90
J = 50 * x(1) + 70 * x(2)
.....
Object Function Jmax = 2900
x(1) = 30
x(2) = 20
```

6.2.2 爬山法

6.2.2.1 爬山法的基本思想

目标函数寻优的问题可以形象地用瞎子爬山来描述。目标函数的极值好比山峰（极大值）或山谷（极小值）。瞎子爬山只能靠手杖前后左右搜索，哪儿最陡，就表示下降或上升最快，这个方向就是前进的方向。对于目标函数的极小值问题就成为寻求最快下降的方向作为搜索方向，一步步逼近最小值。因此爬山法又称最速下降法或一阶梯度法。

以求目标函数 $J = f(X)$ 的极小值为例，说明爬山法的计算步骤。

①设置搜索初值 $X^{(0)}$，计算 $f(X)$。

②搜索次数 i 置。

③计算 $X^{(i)}$ 处的函数 J 的梯度 $G^{(i)}$ 及梯度单位向量 $E^{(i)}$：

$$G^{(i)} = \nabla f(X^{(i)}) \tag{6-26}$$

$$E^{(i)} = G^{(i)} / \| G^{(i)} \| \tag{6-27}$$

式中，G 是梯度向量，$\| G \|$ 是梯度向量的模。其值由 $\| G \| = \sqrt{\sum_j (g_j)^2}$ 决定。

④判别条件 $\| G^{(i)} \| \leqslant \varepsilon_1$ 或 $| f(X^{(i+1)}) - f(X^{(i)}) | \leqslant \varepsilon_2$ 是否满足。若满足，则打印最优解 $X^{(i)}$ 及目标函数值 $f(X^{(i)})$。不满足则在梯度方向的反方向进行搜索。

⑤搜索最优步长 $h^{(i)}$，使

$$f(X^{(i+1)}) = \min_h f(X^{(i)} - hE^{(i)}) \tag{6-28}$$

⑥计算下一个搜索点 $X^{(i)}$

$$X^{(i)} = X^{(i)} - h^{(i)}E^{(i)} \tag{6-29}$$

⑦ $i = i + 1$，转③。

爬山法是非线性规划寻优的一种方法，它具有迭代计算简单、对初值点要求较低、远离极值点时收敛较快等优点，但在接近极值点时收敛速度不快。

6.2.2.2 共轭梯度法的基本思想

爬山法在极值点附近的收敛速度很慢，因此，如何在极值点附近加速收敛就是共轭梯度法解决的问题。共轭梯度法的基本思想是：

①目标函数 $J = f(X)$ 在极值点 X^* 附近展开成泰勒级数，并取到二次导数项。

$$f(X) \doteq f(X^*) + \nabla f(X^*)^T \Delta X + \frac{1}{2}\Delta X^T A \Delta X \tag{6-30}$$

其中，A 是 $f(X)$ 在 X^* 处的二阶偏导数矩阵，即汉森矩阵。而极值点 X^* 处，一阶导数 $\nabla f(X^*)^T \doteq 0$，因此，有

$$f(X) \doteq f(X^*) + \frac{1}{2}\Delta X^T A \Delta X \tag{6-31}$$

即在极值点处目标函数可用二次函数近似。

②二次函数在极值点附近的等高线可近似作为同心的椭圆族处理。

③对于同心的椭圆族，它的任意两根平行的切线的切点连线通过其中心。这个中心就是极值点。切点的连线方向就是共轭方向。

④以共轭方向作为搜索方向，可以直接得到极值点，因此，以共轭方向寻优的方法称为共轭梯度法。

可以证明，对于一个 n 维欧氏空间上非负定的二次函数，可以用不超过 n 次的搜索达到极小值。由于在极值点附近，目标函数可以用二次函数近似，因此，用有限次的迭代计算就能达到目标函数的极小值点。共轭梯度法是具有上述特性的算法，因此收敛速度很快。这种特性称为二次收敛性。

对于 $n \times n$ 维实对称正定矩阵 A，若向量 P 与 Q 满足：

$$P^T A Q = 0 \tag{6-32}$$

则称 P 与 Q 对于矩阵 A 共轭。

下面介绍弗莱彻与里夫斯（Fletcher & Reeves）提出的计算共轭方向的共轭梯度法计算步骤：

①设置搜索初值 $X^{(0)}$ 及精度 ε。

②用一阶梯度方向的反方向作为初始搜索方向，即 $P^{(0)} = -G^{(0)} = -\nabla f(X^{(0)})$。$K$ 置 0。

③沿 $P^{(K)}$ 作一维搜索，求步长 h，使

$$f(X^{(K)} + h^{(K)}P^{(K)}) = \min_h f(X^{(K)} + hP^{(K)}) \tag{6-33}$$

④得到新的点 $X^{(K+1)}$。

$$X^{(K+1)} = X^{(K)} + h^{(K)}P^{(K)} \tag{6-34}$$

⑤判别新点的梯度 $\nabla f(X^{(K+1)}) \leqslant \varepsilon$ 否？如满足，则输出最优解 $X^{(K+1)}$。否则做⑥。

⑥判别是否已进行 n 次迭代，如满足，则 $X^{(o)} \Leftarrow X^{(K+1)}$，做②。否则做⑦。

⑦构造共轭方向 $P^{(K+1)}$。

$$P^{(K+1)} = -\nabla f(X^{(K+1)}) + \beta^{(K)} P^{(K)} \tag{6-35}$$

为满足共轭条件（6－32），应有 $[P^{(K+1)}]^T A P^{(K)} = 0$，

即：$$[-\nabla f(X^{(K+1)}) + \beta^{(K)} P^{(K)}]^T A P^{(K)} = 0$$

$$\therefore \quad \beta^{(K)} = \| \nabla f(X^{(K+1)}) \|^2 / \| \nabla f(X^{(K)}) \|^2 \tag{6-36}$$

⑧$K \Leftarrow K+1$，转③。

计算步骤中，$\| \cdot \|$ 表示·的模。$\nabla f(X)$ 是梯度。计算过程中，每迭代 n 步，就重新计算负梯度方向，这样处理可避免舍入误差的积累造成不收敛，同时，第一次搜索方向是负梯度方向，有利于突破目标函数的非二次性。

6.2.2.3　共轭梯度法仿真程序

爬山法和共轭梯度法用于求解无约束条件的非线性规划问题。下面给出了共轭梯度法的仿真程序。程序中变量说明如下：

N：变量维数。

E：允许误差，即精度。

X（I）：初始点坐标。

F：目标函数的值。

G（I）：梯度值 $\nabla f(X)$

S1：$\nabla f(X^{(K+1)})$ 的模。

S0：$\nabla f(X^{(K)})$ 的模。

T0：步长搜索的外推内插法的初始步长。

V：迭代时步长缩短的倍数。

程序运行前要完成下列准备工作。

①输入目标函数表达式。

在2030语句输入目标函数表达式。例如目标函数是 $J = 4x_1^2 + x_2^2 - 40x_1 - 12x_2 + 136$，则输入语句是：

```
2030  F=4*X(1)^2+X(2)^2-40*X(1)-12*X(2)+136
```

②输入各变量的梯度表达式。

从2500语句开始送入根据目标函数求出的各变量的梯度表达式。如上例中，有

$$G(1) = \frac{\partial f}{\partial x_1} = 8x_1 - 40; G(2) = \frac{\partial f}{\partial x_2} = 2x_2 - 12$$

因此，从2500语句开始送入下列子程序。

```
2500  G(1)=8*X(1)-40
2510  G(2)=2*X(2)-12
2520  RETURN
```

最后，要送入RETURN语句，作为子程序结束。下面给出的程序就是用于求解上述问题的，对其他寻优问题可按上述方法送入有关语句。对于求极大值的问题，只要把目标函数取负值，即成为求极小值的问题，并可用本程序求解。

程序经做了上述准备工作后，即可运行，先回答变量数，本例为2个变量，故打入2并回车，再回答精度，如打入0.00001并回车，最后送入两个变量的初值，经回车后，打印机打印出下列结果：

```
Object Function J=0
```

x(1) = 5.000001

x(2) = 5.999995

即极小值为0。最优解 $x_1=5; x_2=6$。

共轭梯度法仿真程序清单如下：

```
1000  CLS
1005  REM Fletcher & Reeves Method to Solve Non - LP
1010  LOCATE 2, 1
1015  PRINT"F - P Method to Solve Non - LP Problems"
1020  REM N: Variable Number E: Allowable Error
1025  LOCATE 4, 1
1030  INPUT "Variable Number"; N
1035  LOCATE 6, 1
1040  INPUT "Allowable Error"; E
1045  DIM X(N), Y(N), G(N), P(N)
1050  LOCATE 8, 1
1055  PRINT "Input Start Value of The Variables"
1060  FOR I = 1 TO N
1065  PRINT "x("; I; ") = "; : INPUT X(I)
1070  NEXT I
1075  PRINT
1080  FOR K = 1 TO N
1085  GOSUB 2500
1090  S1 = 0
1095  FOR I = 1 TO N
1100  Y(I) = X(I)
1105  S1 = G(I) * G(I) + S1
1110  NEXT I
1115  IF S1 < E THEN 1510
1120  FOR I = 1 TO N
1125  IF K < > 1 THEN 1140
1130  P(I) = - G(I)
1135  GOTO 1145
1140  P(I) = - G(I) + S1/S0 * P(I)
1145  NEXT I
1150  T0 = 1
1155  V = .1
1160  H1 = 0
1165  H = H1
1170  GOSUB 2000
1175  F1 = F
```

```
1180  U = 0
1185  T = T0
1190  H2 = H1 + T
1195  H = H2
1200  GOSUB 2000
1205  F2 = F
1210  IF F1 > F2 THEN 1255
1215  T = - T
1220  H3 = H1
1225  F3 = F1
1230  H1 = H2
1235  F1 = F2
1240  H2 = H3
1245  F2 = F3
1250  GOTO 1265
1255  T = T + T
1260  U = U + 1
1265  H3 = H2 + T
1270  H = H3
1275  GOSUB 2000
1280  F3 = F
1285  IF F2 > F3 THEN 1310
1290  IF U > 0 THEN 1325
1295  C1 = (F3 - F1)/(H3 - H1)
1300  C2 = ((F2 - F1)/(H2 - H1) - C1)/(H2 - H3)
1305  GOTO 1390
1310  T = T + T
1315  U = U + 1
1320  GOTO 1230
1325  H4 = .5 * (H2 + H3)
1330  H = H4
1335  GOSUB 2000
1340  F4 = F
1345  IF F4 > F2 THEN 1375
1350  H1 = H2
1355  F1 = F2
1360  H2 = H4
1365  F2 = F4
1370  GOTO 1295
1375  H3 = H4
```

```
1380  F3 = F4
1385  GOTO 1295
1390  IF ABS (C2) < E THEN 1465
1395  H4 = .5 * (H1 + H3 - C1/C2)
1400  H = H4
1405  GOSUB 2000
1410  F4 = F
1415  IF F2 < 1 THEN 1430
1420  F5 = F2
1425  GOTO 1435
1430  F5 = 1
1435  IF ABS ((F4 - F2)/F5) < E THEN 1480
1440  IF F4 > F2 THEN 1465
1445  H1 = H4
1450  F1 = F4
1455  T0 = T0 * V
1460  GOTO 1180
1465  H1 = H2
1470  F1 = F2
1475  GOTO 1455
1480  FOR I = 1 TO N
1485  X(I) = Y(I) + H4 * P(I)
1490  NEXT I
1495  S0 = S1
1500  NEXT K
1505  GOTO 1080
1510  H = 0
1515  GOSUB 2000
1520  PRINT "Object Function J = "; F
1525  FOR I = 1 TO N
1530  PRINT "x("; I; ") = "; X(I)
1535  NEXT I
1540  END
2000  FOR I = 1 TO N
2010  X(I) = Y(I) + H * P(I)
2020  NEXT I
2030  F = 4 * X(I)^2 + X(2)^2 - 40 * X(1) - 12 * X(2) + 136
2040  RETURN
2500  G(1) = 8 * X(1) - 40
2510  G(2) = 2 * X(2) - 12
```

2520 RETURN

〔例 6－2〕求解下列目标函数的极小值。

$$J = x_1^2 + x_2^2 - x_1x_2 - 10x_1 - 4x_2 + 60$$

输入下列语句：

```
2030 F=X(1)^2+X(2)^2-X(1)*X(2)-10*X(1)-4*X(2)+60
2500 G(1)=2*X(1)-X(2)-10
2510 G(2)=2*X(2)-X(1)-4
2520 RETURN
```

运行后输入变量数 2 及精度、初值后，打印结果是：

```
Object Function J=8
x(1)=8
x(2)=6
```

6.2.3 优选法

优选法又称黄金分割法、0.618 法，它是通过不断缩短单峰区间的长度来搜索极值的方法。搜索区间按比例 $\lambda=0.618$ 缩小，通过计算和比较 $f(x)$ 的函数值，确定取舍的区间。

设搜索区间的长度为 L，它分割成两部分。长的一段区间为 x，短的一段则为 $L-x$，它满足

$$\frac{L}{x} = \frac{x}{L-x} = \frac{1}{\lambda} \tag{6-37}$$

这种分割称为黄金分割。由(6－37)可以求得

$$\lambda = \frac{-1+\sqrt{5}}{2} = 0.6180339887\cdots$$

由此，本法又称 0.618 法。其计算步骤如下：

①设置搜索区间，设为$[a_0, b_0]$。

②$A \Leftarrow a_0, B \Leftarrow b_0$

③计算中间点 x_1、x_2 及其函数值 F_1、F_2：

$$x_1 = A + \lambda(B-A); x_2 = B - \lambda(B-A) \tag{6-38}$$

$$F_1 = f(x_1); F_2 = f(x_2) \tag{6-39}$$

式中，$\lambda=0.618$。$f(x)$ 是目标函数值。优选 法是单变量的非线性规划寻优方法。因此，变量只有一个。

④比较 F_1 与 F_2 的大小，如 $F_1 \geqslant F_2$ 则进行下列替换，以缩短搜索区间。

$$B = x_1; x_1 = x_2; x_2 = B - \lambda(B-A); F_1 = F_2; F_2 = f(x_2)$$

并转⑥，如不满足，则转⑤。

⑤进行下列替换，以缩短搜索区间，计算后转⑥：

$$A = x_2; x_2 = x_1; x_1 = A + \lambda(B-A); F_2 = F_1; F_1 = f(x_1)$$

⑥判别搜索区间是否满足要求的精度，即：

$$\left|\frac{B-A}{b_0-a_0}\right| \leqslant \delta$$

如满足则做⑦，否则做④。

⑦比较 F_1 与 F_2 的大小，打印出小的那个函数值与相应的区间值。

黄金分割法的优点是每次消去部分区间后，只要再计算一个对称点（原保留点的对称点）的函数值就能进行下次的比较，因此，计算工作量较少。而且区间的缩短率不变，为0.618，所以计算 n 个点后，总缩短率是 λ^{n-1}。收缩速度较快。

黄金分割法寻优的方法比较简单，仿真程序不另给出。

6.2.4 单纯形加速法

对于多变量函数的寻优问题，可以采用爬山法和共轭梯度法求解。但是这两种方法都要求计算目标函数的梯度，对于实际问题，由于目标函数的解析表达式不一定能获得，目标函数的梯度只能由计算机近似计算给出，因此，影响了寻优的收敛性，增加了计算的工作量。单纯形加速法是一种不必计算目标函数梯度的直接搜索法。它计算若干点的函数值，比较它们的大小，判别函数的变化趋势，确定搜索方向。因此，本方法操作简便，易于实施，得到了广泛的应用。

6.2.4.1 单纯形加速法的基本思想

单纯形是在一定空间中最基本最简单的图形。在二维空间，单纯形是三角形，三维空间中的单纯形是四面体。对于 n 维空间，单纯形是以 $n+1$ 个顶点组成的图形。如 $n+1$ 个顶点的距离相等，则称为正单纯形，如二维空间中的正三角形，三维空间的正四面体。

单纯形加速法是按一定的模式进行目标函数最优点的搜索的。首先是构造单纯形，其次是确定反射点，进行扩张和回缩，然后是单纯形收缩，直到达到极值点。下面以二元函数为例，说明单纯形法是如何搜索的。

设目标函数 $J=f(x_1,x_2)$，先取不在一直线上的三个点 p_G、p_H、p_L 组成单纯形，如图6－1所示。算出三个顶点的目标函数值 f_G、f_H、f_L，对它们比较的结果，若有 $f_H>f_G>f_L$，则对于求极小值的问题来说，f_H 的对应点 p_H 必定是坏点，考虑函数变化的趋势，好点应在坏点的对称点 p_R。

p_R 是通过连接 p_G 与 p_L 线段中点 p_F 与 p_H，并延长到 p_R，使 p_Fp_R 线段长度等于 p_Hp_F 线段长度而求得，如图6－1所示。这是确定反射点的过程。

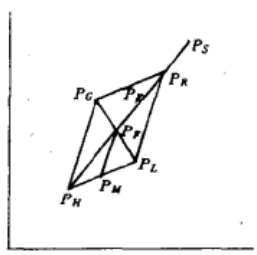


图6－1 单纯形加速法求反射点

然后，按一定的扩张因子或压缩因子对 p_R 进行扩张或压缩，如达到图6－1中的 p_E 或 p_S 点，使得新点 p_R 的值 $f_R<f_G$，则 f_G 相应点成为下一次迭代的坏点。这是反射点的扩张和回缩过程。

如果通过扩张和回缩，找不到新点 p_R 使得 $f_R<f_G$，则应对原来的单纯形按一定比例缩小，组成新的单纯形，如缩小边长一半，组成由 p_L、p_F 与 p_M 的新的单纯形，如图6－1所示。然后重新进行比较。直到满足所需的收敛条件为止。这个过程是单纯形的收缩。

6.2.4.2 单纯形加速法的计算步骤

单纯形加速法的计算步骤及有关计算公式说明如下：

①给出目标函数 $J=f(X)$ 的表达式及 X 的初值 X_0 和变量个数 n，精度 E，迭代次数 K_1 置0。

②确定单纯形顶点 X_i 的坐标：

$$X_i = X_0 + hE_i \tag{6-40}$$

E_i 是第 i 个单位座标向量。h 是初始步长。

③计算各顶点 X_i 的目标函数值 $S_i = f(X_i)$。

④比较 S_i，求出最大值 S_H，次大值 S_G 及最小值 S_L，并确定相应的 X_H、X_G 及 X_L。

⑤比较 $\frac{|S_H - S_L|}{|S_L|} < E$ 否，若满足，则打印 X_L，S_L。否则迭代次数加 1，若次数超过设定值则表示搜索失败。若不超过则转⑥。

⑥确定反射点 X_R，计算 S_R：

$$X_R = \frac{2}{n}\left(\sum_{i=0}^{n} X_i - X_H\right) - X_H = 2X_F - X_H \tag{6-41}$$

⑦判别 $S_R < S_G$ 否？若不满足则进行回缩，计算新点 X_S 及 S_S。

$$X_R = (1-\lambda)X_H + \lambda X_R; S_S = f(X_S) \tag{6-42}$$

再判别 $S_S < S_G$ 否？若满足，转⑧，否则转⑨。

若满足 $S_R < S_G$，则进行扩张的判别：

$$(1+\mu)S_H + \mu S_R < S_L \tag{6-43}$$

若满足扩张条件，则求新点 X_E 及 S_E：

$$X_E = (1-\mu)X_H + \mu X_R; S_E = f(X_E) \tag{6-44}$$

再判别 $S_E < S_R$ 否？满足则有

$X_0 = X_E$；$S_0 = S_E$。转②。

不满足则有 $X_S = X_R$；$S_S = S_R$。转⑧

⑧新的坏点 $X_H = X_S$ 及 $S_H = S_S$。转④。

⑨进行收缩边长，新的顶点 X_i 是

$$X_i = \frac{1}{2}(X_L + X_i) \tag{6-45}$$

并转③。

计算步骤中，μ 是扩张因子，λ 是回缩因子。

6.2.4.3 单纯形加速法仿真程序

由于单纯形加速法不必知道目标函数的解析表达式，因此，在实际应用中使用范围很广。单纯形加速法仿真程序中的变量说明如下：

N：变量维数。

T：回缩因子。本程序取 0.75。

U：扩张因子。本程序取 2。

L_0：初始步长。本程序取 2。根据经验，在开始时，可取 1.6～1.7。进行一定阶段后，如发现 $S_H - S_L$ 已很小，但未达精度要求，这时，可取好点 X_0，以初始步长 0.5～1 重新开始。如发现远离精度要求，可取初始步长大些，如 2～15 间的值，然后从好点开始重新搜索。

X(I,J)：X_i 坐标。($J = 0, \cdots, N$)

X(I,N+1)，X(I,N+2)：X_R，X_S 或 X_E 的坐标。

程序需要在运行前做下列准备工作。

①输入目标函数计算式。

从 2000 语句开始送入目标函数计算式。如目标函数 $J = x_1^2 + x_2^2 - x_1x_2$ 则应送入

```
2000  S(J) = X(1,J)^2 + X(2,J)^2 - X(1,J) * X(2,J)
2010  RETURN
```

②对输出结果进行处理。

对于目标函数寻优，其最优解在 $X(I,J2)$ 内($I=1,2,\cdots,n$)。目标函数极值在 $Q(J2)$ 内。当用于其他优化计算时，对这部分程序应做修改。

例如，在控制系统的对象特性测试时，常在对象输入端加入阶跃信号，测取输出端的响应曲线。若对象用 $G(s)=\frac{Ke^{-\tau s}}{Ts+1}$ 来近似，则可用单纯形加速法对所测数值寻优，找到最佳的 K、T、τ 值。其目标函数可取计算值与实测值差的平方和。实测值与对应的时间值应编制有关程序输入。

程序中的扩张因子 U、回缩因子 T 及初始步长也可以改变为其他值，这只要改变 1010 至 1020 语句即可。

程序运行时，输入变量个数 N 及 N 个状态初值后，就会打印出最优解时的目标函数值及最优解。在 CRT 上显示的:与·表示进行扩张及回缩。由于寻优迭代过程较长，因此，显示:与 . 还可以表示运算正在进行中。

下面给出单纯形加速法的仿真程序清单。

```
1000  REM Simplex Method to Solve Optimal Problem
1005  READ N:REM The Number or The Variables
1010  LO=1.7
1015  T=.75
1020  U=2
1025  E=.0001
1030  DIM X(N,N+3),S(N+3),Q(N+3)
1035  CLS
1040  FOR I=1 TO N
1045  READ X(I,0):REM The Start Value of Variables
1050  NEXT I
1055  PRINT
1060  FOR J=1 TO N
1065  FOR I=1 TO N
1070  IF I=J THEN X(I,J)=X(I,0)+LO
1075  IF I<>J THEN X(I,J)=X(I,0)
1080  NEXT I
1085  NEXT J
1090  FOR J=0 TO N
1095  GOSUB 2000
1100  Q(J)=S(J)
1105  NEXT J
1110  J0=0
1115  J2=0
1120  FOR J=1 TO N
1125  IF Q(J)>Q(J0) THEN J0=J
```

```
1130  IF Q(J)>=Q(J2) THEN 1140
1135  J2=J
1140  NEXT J
1145  J1=J2
1150  FOR J=0 TO N
1155  IF J=J0 THEN 1170
1160  IF Q(J)<=Q(J1) THEN 1170
1165  J1=J
1170  NEXT J
1175  Q0=Q(J0)
1180  Q1=Q(J1)
1185  Q2=Q(J2)
1190  FOR I=1 TO N
1195  FOR J=1 TO N
1200  IF ABS(X(I,J)-X(I,0))>E THEN 1235
1205  NEXT J
1210  NEXT I
1215  IF Q2>1 THEN 1230
1220  IF (Q0-Q2)<E THEN 1515
1225  GOTO 1235
1230  IF (Q0-Q2)<(E*Q2) THEN 1515
1235  FOR I=1 TO N
1240  X(I,N+1)=0
1245  FOR J=0 TO N
1250  X(I,N+1)=X(I,N+1)+X(I,J)
1255  NEXT J
1260  X(I,N+1)=2*(X(I,N+1)-X(I,J0))/N-X(I,J0)
1265  NEXT I
1270  J=N+1
1275  GOSUB 2000
1280  Q(N+1)=S(J)
1285  IF Q(N+1)<Q(J1) THEN 1420
1290  IF Q(N+1)<Q(J0) THEN 1320
1295  FOR I=1 TO N
1300  X(I,N+2)=X(I,N+1)
1305  X(I,N+1)=X(I,J0)
1310  X(I,J0)=X(I,N+2)
1315  NEXT I
1320  FOR I=1 TO N
1325  X(I,N+2)=(1-T)*X(I,J0)+T*X(I,N+1)
```

```
1330  NEXT I
1335  J=N+2
1340  GOSUB 2000
1345  Q(N+2)=S(J)
1350  IF Q(N+2)<Q(J1) THEN 1390
1355  FOR J=0 TO N
1360  FOR I=1 TO N
1365  X(I,J)=(X(I,J)+X(I,J2))/2
1370  NEXT I
1375  NEXT J
1380  PRINT ".";
1385  GOTO 1090
1390  FOR I=1 TO N
1395  X(I,J0)=X(I,N+2)
1400  MEXT I
1405  Q(J0)=Q(N+2)
1410  PRINT ":";
1415  GOTO 1110
1420  IF (1-U)*Q0+U*Q(N+1)<Q2 THEN 1450
1425  FOR I=1 TO N
1430  X(I,N+2)=X(I,N+1)
1435  NEXT I
1440  Q(N+2)=Q(N+1)
1445  GOTO 1390
1450  FOR I=1 TO N
1455  X(I,N+3)=(1-U)*X(I,J0)+U*X(I,N+1)
1460  NEXT I
1465  J=N+3
1470  GOSUB 2000
1475  Q(N+3)=S(J)
1480  IF Q(N+3)<Q(N+1) THEN 1490
1485  GOTO 1425
1490  FOR I=1 TO N
1495  X(I,0)=X(I,N+3)
1500  NEXT I
1505  GOTO 1060
1510  CLS
1515  REM Print Optimal Solution
1520  LPRINT
1525  LPRINT "Object Function Jmin=";Q(J2)
```

```
1530  PRINT
1535  PRINT "Object Function Jmin=";Q(J2)
1540  FOR I=1 TO N
1545  PRINT "x(";I;")=";INT(X(I,J2)*1000)/1000
1550  LPRINT "x(";I;")=";INT(X(I,J2)*1000)/1000
1555  NEXT I
1560  END
2000  REM Object Function Express
2005  S(J)=X(1,J)^2+X(2,J)^2-X(1,J)*X(2,J)-10*X(1,J)-4*X(2,J)+60
2010  RETURN
```

程序清单中的目标函数是2000～2010语句表示的。

$$J = x_1^2 + x_2^2 - x_1 \cdot x_2 - 10x_1 - 4x_2 + 60$$

运行时，还要先送入变量数2及相应的初值，用DATA语句，如：

```
5000  DATA 2, 0, 0
```

其中，2表示有两个变量，即 x_1 及 x_2。两个0表示初值为0。

运行时，有一系列：或．出现，表明计算机正在搜索。最后的结果，在CRT与打印机同时输出。下面是打印的结果：

```
Object Function Jmin=8
x (1) =7.999
x (2) =6.001
```

〔例6－3〕已知被测对象的阶跃响应，确定对象的 K、T、τ 值。

经修改后的程序清单如下：

```
1000  REM Simplex Method to Solve Optimal Problem
1005  READ N:REM The Number of The Parameters
1010  LO=1.7
1015  T=.75
1020  U=2
1025  E=.01:UI=.1
1030  READ TN:REM TN:The Number of Time Points
1035  DIM X(N,N+3),S(N+3),Q(N+3),YC(TN),YT(TN),TI(TN)
1040  FOR I=1 TO TN
1045  READ TI(I),YT(I):REM YT(I):Response in TI(I)
1050  NEXT
1055  CLS
1060  LOCATE 2,1
1065  PRINT "Input The Start Value:"
1070  LOCATE 4,1
1075  INPUT "The Gain Ko=";X(1,0)
1080  LOCATE 6,1
1085  INPUT "The Time Constant To=";X(2,0)
```

```
1090  LOCATE 8,1
1095  INPUT "The Time Delay Tor=";X(3,0)
1100  PRINT
1105  FOR J=1 TO N
1110  FOR I=1 TO N
1115  IF I=J THEN X(I,J)=X(I,0)+LO
1120  IF I<>J THEN X(I,J)=X(I,0)
1125  NEXT I
1130  NEXT J
1135  FOR J=0 TO N
1140  GOSUB 2000
1145  Q(J)=S(J)
1150  NEXT J
1155  J0=0
1160  J2=0
1165  FOR J=1 TO N
1170  IF Q(J)>Q(J0) THEX J0=J
1175  IF Q(J)>=Q(J2) THEN 1185
1180  J2=J
1185  NEXT J
1190  J1=J2
1195  FOR J=0 TO N
1200  IF J=J0 THEN 1215
1205  IF Q(J)<=Q(J1) THEN 1215
1210  J1=J
1215  NEXT J
1220  Q0=Q(J0)
1225  Q1=Q(J1)
1230  Q2=Q(J2)
1235  FOR I=1 TO N
1240  FOR J=1 TO N
1245  IF ABS(X(I,J)-X(I,0))>E THEN 1280
1250  NEXT J
1255  NEXT I
1260  IF Q2>1 THEN 1275
1265  IF (Q0-Q2)<E THEN 1560
1270  GOTO 1280
1275  IF (Q0-Q2)<(E*Q2) THEN 1560
1280  FOR I=1 TO N
1285  X(I,N+1)=0
```

```
1290  FOR J=0 TO N
1295  X(I,N+1)=X(I,N+1)+X(I,J)
1300  NEXT J
1305  X(I,N+1)=2*(X(I,N+1)-X(I,J0))/N-X(I,J0)
1310  NEXT I
1315  J=N+1
1320  GOSUB 2000
1325  Q(N+1)=S(J)
1330  IF Q(N+1)<Q(J1) THEN 1465
1335  IF Q(N+1)<Q(J0) THEN 1365
1340  FOR I=1 TO N
1345  X(I,N+2)=X(I,N+1)
1350  X(I,N+1)=X(I,J0)
1355  X(I,J0)=X(I,N+2)
1360  NEXT I
1365  FOR I=1 TO N
1370  X(I,N+2)=(1-T)*X(I,J0)+T*X(I,N+1)
1375  NEXT I
1380  J=N+2
1385  GOSUB 2000
1390  Q(N+2)=S(J)
1395  IF Q(N+2)<Q(J1) THEN 1435
1400  FOR J=0 TO N
1405  FOR I=1 TO N
1410  X(I,J)=(X(I,J)+X(I,J2))/2
1415  NEXT I
1420  NEXT J
1425  PRINT ".";
1430  GOTO 1135
1435  FOR I=1 TO N
1440  X(I,J0)=X(I,N+2)
1445  NEXT I
1450  Q(J0)=Q(N+2)
1455  PRINT":";
1460  GOTO 1155
1465  IF (1-U)*Q0+U*Q(N+1)<Q2 THEN 1495
1470  FOR I=1 TO N
1475  X(I,N+2)=X(I,N+1)
1480  NEXT I
1485  Q(N+2)=Q(N+1)
```

```
1490  GOTO 1435
1495  FOR I=1 TO N
1500  X(I,N+3)=(1-U)*X(I,J0)+U*X(I,N+1)
1505  NEXT I
1510  J=N+3
1515  GOSUB 2000
1520  Q(N+3)=S(J)
1525  IF Q(N+3)<Q(N+1) THEN 1535
1530  GOTO 1470
1535  FOR I=1 TO N
1540  X(I,0)=X(I,N+3)
1545  NEXT I
1550  GOTO 1105
1555  CLS
1560  LPRINT "TIME Yin Ycal"
1565  LPRINT "(Minute)"
1570  PRINT "TIME Yin Ycal"
1575  PRINT "(Minute)"
1580  FOR I=0 TO TN
1585  REM S=I*TP
1590  S=TI(I)
1595  PRINT "   ";S;SPC(20-LEN(STR$(S)));YT(I);
1600  PRINT SPC(20-LEN(STR$(YT(I))));YC(I)
1605  LPRINT "   ";S;SPC(20-LEN(STR$(S)));YT(I);
1610  LPRINT SPC(20-LEN(STR$(YT(I))));YC(I)
1615  NEXT I
1620  LPRINT
1625  PRINT
1630  PRINT"Process Gain";INT(X(1,J2)*100)/100
1635  PRINT "Time Constant";INT(X(2,J2)*100)/100;"min"
1640  PRINT "Time Delay ";INT(X(3,J2)*100)/100;"min"
1645  LPRINT SPC(8);"Process Gain ";INT(X(1,J2)*100)/100
1650  A2=INT(X(2,J2)*100)/100
1655  LPRINT SPC(8);"Time Delay ";A2;SPC(8-LEN(STR$(A2)));"min."
1660  A3=INT(X(3,J2)*100)/100
1665  LPRINT SPC(8);"Time Delay ";A3;SPC(8-LEN(STR$(A3)));"min."
1670  END
2000  S(J)=0
2005  FOR I=0 TO TN
2010  REM S=I*TP
```

```
2015  S=TI(I)
2020  IF S<X(3,J)THEN YC(I)=0:GOTO 2035
2025  IF X(3,J)<0 THEN X(3,J)=0
2030  YC(I)=X(1,J)*(1-EXP(-(S-X(3,J))/X(2,J)))*UI
2035  S(J)=S(J)+(YC(I)-YT(I))^2
2040  NEXT I
2045  RETURN
```

在运行前要输入相应的数据，其次序是：变量数，点数，时间，相应时间的输出值，时间，相应时间的输出值，…，直到各点数的时间及输出数据全部输入。本例中有 38 个点，数据如下：

```
5000  DATA 3,38,1,6E-3,2,.017,3,.025,4,.032,5,.039,6,.044
5005  DATA 7,.049,8,.055,9,.06,10,.064,11,.069,12,.073
5010  DATA 13,0.077,14,.081,15,.083,16,.086,17,.089,18,0.092
5015  DATA 19,.095,20,.097,21,.1,22,.1 04,23,.107,24,.11
5020  DATA 25,.113,26,.114,27,.115,28,.116,29,.118,30,.12
5025  DATA 31,.122,32,.124,33,.125,34,.126,35,.127
5030  DATA 36,.128,37,.13,38,.131
```

数据中，3 表示有 3 个变量要估计参数值，即 K、T、τ 值。38 是点数。1，2，…，38，是相应的时间。其单位是分。0.006，…，0.131 是输出值。阶跃信号 UI 为 0.1。

运行时，输入相应 K、T、τ 的初值，即进行搜索。最后结果同时在 CRT 与打印机输出。本例的结果打印如下：

TIME (Minute)	Yin	Ycal
0	0	0
1	.006	4.250567E-03
2	.017	1.324533E-02
3	.025	.0216172
4	.032	2.940934E-02
5	.039	3.666186E-02
6	.044	4.341217E-02
7	.049	4.969502E-02
8	.055	5.554277E-02
9	.06	6.098559E-02
10	.064	6.605149E-02
11	.069	7.076658E-02
12	.073	7.515516E-02
13	.077	7.923983E-02
14	.081	8.304163E-02
15	.083	8.658018E-02
16	.086	8.987366E-02

17	.089	9.293909E－02
18	.092	9.579222E－02
19	.095	9.844779E－02
20	.097	.1009195
21	.1	.10322
22	.104	.1053612
23	.107	.1073541
24	.11	.109209
25	.113	.1109355
26	.114	.1125424
27	.115	.114038
28	.116	.1154301
29	.118	.1167257
30	.12	.1179317
31	.122	.1190541
32	.124	.1200988
33	.125	.1210712
34	.126	.1219762
35	.127	.1228185
36	.128	.1236026
37	.13	.1243323
38	.131	.1250115

Process Gain 1.34

Time Constant 13.91 min.

Time Delay .56 min.

可以看到，经搜索，过程的 K、T、τ 均已在结果中显示。应该指出，本程序中的精度 $E=0.01$，因此，对不同的初值，其估计的值会有所不同，通过对某些参数加入权重函数，减小 E 的值等措施，可以提高参数估计的精度。

6.2.5 拉格朗日（Lagrange）乘子法

对于具有约束条件的非线性函数寻优问题，目前还没有一种十分有效的方法。拉格朗日乘子法是较为有效的一种寻优算法，它可以用于等式约束条件的多变量函数寻优，也适用于不等式约束条件的多变量函数寻优。在动态最优的各种方法中，拉格朗日乘子法也是一种可行的方法。

6.2.5.1 拉格朗日乘子法的基本思想

按照线性规划法中介绍的步骤，对于不等式约束条件，可以通过引入松弛变量，使不等式约束条件成为等式约束条件。与线性规划中不同的是，各变量没有非负的约束，即不要求满足 $x_i \geqslant 0$（$i=1, \cdots, n$），因此，为使不等式成立，在拉格朗日乘子法中，引入的松弛变量均用平方项，以保证该引入项为非负。

拉格朗日乘子法的基本思想是对于等式约束条件

$$g(x) = 0 \ ; x = [x_1, x_2, \cdots, x_n]^T ; g = [g_1(x), g_2(x), \cdots, g_m(x)]^T ; m < n \tag{6-46}$$

的目标函数 $J = f(x)$ 的寻优问题，通过引入拉格朗日乘子向量 λ，使其成为求解无约束条件下拉格朗日函数 $L(x,\lambda)$

$$L(x,\lambda) = f(x) + \lambda^T g(x) \tag{6-47}$$

的求极值问题。

根据对目标函数求极值的必要条件，可以得到具有等式约束条件时函数 $f(x)$ 的极值存在的必要条件是拉格朗日函数 $L(x,\lambda)$ 的偏导数为0。即

$$\begin{cases} \dfrac{\partial L(x,\lambda)}{\partial x} = 0 \\ \dfrac{\partial L(x,\lambda)}{\partial \lambda} = 0 \end{cases} \tag{6-48}$$

上述 $n+m$ 个标量方程组的解 x 即为最优解。解 λ 即为待定乘子的值，其物理意义是极值附近，目标函数 $f(x)$ 随约束条件 $g(x)$ 微小变化而变化的比率。

〔例6-4〕求等式约束条件 $g(x) = x_1 + x_2 - 8 = 0$ 时，目标函数 $J = 60 - 10x_1 - 4x_2 + x_1^2 + x_2^2 - x_1x_2$ 的极小值。

列出拉格朗日函数 $L(x,\lambda)$：

$$L(x,\lambda) = 60 - 10x_1 - 4x_2 + x_1^2 + x_2^2 - x_1x_2 + \lambda(x_1 + x_2 - 8)$$

$$\therefore \quad \frac{\partial L}{\partial x_1} = -10 + 2x_1 - x_2 + \lambda = 0$$

$$\frac{\partial L}{\partial x_2} = -4 + 2x_2 - x_1 + \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = x_1 + x_2 - 8 = 0$$

其解为 $x_1=5$；$x_2=3$；$\lambda=3$；$f_{\min}=17$

6.2.5.2 拉格朗日乘子法仿真程序

为便于在计算机上直接利用寻优算法求解极值，通常，引入新的目标函数 $S(J)$

$$S(J) = \sum_{i=1}^{n}\left(\frac{\partial L}{\partial x_i}\right)^2 + \sum_{k=1}^{m}[g_k(x)]^2 \tag{6-49}$$

这是由于对于 $n+m$ 个变量的 $n+m$ 个有偏导的方程组，求解比较繁杂，而直接通过寻优法可以较快得到最优解。

直接寻优的算法很多，下面给出的是用单纯形法寻优的拉格朗日乘子法仿真程序。程序中各变量和数组说明如下：

T、U、LO：单纯形法中的回缩扩张、因子及初始步长。通过1015～1025语句可以修改其数值。

E：精度。

N0：变量个数。

M0：约束条件个数。

X(I,0)：变量初值。包括松弛变量的初值。

F：目标函数值。

W(I)：约束条件表达式。对于不等式约束条件，引入松弛变量的平方项。

G(I)：目标函数的一阶偏导表达式。

D(I,K)：等式约束条件 W(I) 对第 K 个变量 X(K,J) 的一阶偏导表达式。

打印结果中，X 表示最优解，L 表示拉格朗日乘子。J_{min}表示目标函数极小值。

拉格朗日乘子法仿真程序清单如下：

```
1000  REM Langrange Method to Solve Optimal Problem
1005  READ NO:REM The Number of The Variables
1010  READ MO:REM The Number of The Constract
1015  LO=1.7
1020  T=.75
1025  U=2
1030  E=.0001
1035  N=NO+MO
1040  DIM X(N,N+3),S(N+3),Q(N+3),G(N0),W(M0),D(M0,N0)
1045  CLS
1050  FOR I=1 TO N
1055  READ X(I,0):REM The Start Value of Variables
1060  NEXT I
1065  PRINT
1070  FOR J=1 TO N
1075  FOR I=1 TO N
1080  IF I=J THEN X(I,J)=X(I,0)+LO
1085  IF I<>J THEN X(I,J)=X(I,0)
1090  NEXT I
1095  NEXT J
1100  FOR J=0 TO N
1105  GOSUB 3000
1110  Q(J)=S(J)
1115  NEXT J
1120  J0=0
1125  J2=0
1130  FOR J=1 TO N
1135  IF Q(J)>Q(J0) THEN J0=J
1140  IF Q(J)>=Q(J2) THEN 1150
1145  J2=J
1150  NEXT J
1155  J1=J2
1160  FOR J=0 TO N
1165  IF J=J0 THEN 1180
1170  IF Q(J)<=Q(J1) THEN 1180
1175  J1=J
1180  NEXT J
1185  Q0=Q(J0)
```

```
1190  Q1 = Q(J1)
1195  Q2 = Q(J2)
1200  FOR I = 1 TO N
1205  FOR J = 1 TO N
1210  IF ABS(X(I, J) - X(I, 0)) > E THEN 1245
1215  NEXT J
1220  NEXT I
1225  IF Q2 > 1 THEN 1240
1230  IF (Q0 - Q2) < E THEN 1525
1235  GOTO 1245
1240  IF (Q0 - Q2) < (E * Q2) THEN 1525
1245  FOR I = 1 TO N
1250  X(I, N + 1) = 0
1255  FOR J = 0 TO N
1260  X(I, N + 1) = X(I, N + 1) + X(I, J)
1265  NEXT J
1270  X(I, N + 1) = 2 * (X(I, N + 1) - X(I, J0))/N - X(I, J0)
1275  NEXT I
1280  J = N + 1
1285  GOSUB 3000
1290  Q(N + 1) = S(J)
1295  IF Q(N + 1) < Q(J1) THEN 1430
1300  IF Q(N + 1) < Q(J0) THEN 1330
1305  FOR I = 1 TO N
1310  X(I, N + 2) = X(I, N + 1)
1315  X(I, N + 1) = X(I, J0)
1320  X(I, J0) = X(I, N + 2)
1325  NEXT I
1330  FOR I = 1 TO N
1335  X(I, N + 2) = (1 - T) * X(I, J0) + T * X(I, N + 1)
1340  NEXT I
1345  J = N + 2
1350  GOSUB 3000
1355  Q(N + 2) = S(J)
1360  IF Q(N + 2) < Q(J1) THEN 1400
1365  FOR J = 0 TO N
1370  FOR I = 1 TO N
1375  X(I, J) = (X(I, J) + X(I, J2))/2
1380  NEXT I
1385  NEXT J
```

```
1390  PRINT ".";
1395  GOTO 1100
1400  FOR I=1 TO N
1405  X(I, J0)=X(I, N+2)
1410  NEXT I
1415  Q(J0)=Q(N+2)
1420  PRINT":";
1425  GOTO 1120
1430  IF (1-U)*Q0+U*Q(N+1)<Q2 THEN 1460
1435  FOR I=1 TO N
1440  X(I, N+2)=X(I, N+1)
1445  NEXT I
1450  Q(N+2)=Q(N+1)
1455  GOTO 1400
1460  FOR I=1 TO N
1465  X(I, N+3)=(1-U)*X(I, J0)+U*X(I, N+1)
1470  NEXT I
1475  J=N+3
1480  GOSUB 3000
1485  Q(N+3)=S(J)
1490  IF Q(N+3)<Q(N+1) THEN 1500
1495  GOTO 1435
1500  FOR I=1 TO N
1505  X(I, 0)=X(I, N+3)
1510  NEXT I
1515  GOTO 1070
1520  CLS
1525  REM Print Optimal Solution
1530  LPRINT
1535  PRINT
1540  FOR I=1 TO NO
1545  PRINT"X(";I;")=";INT(X(I, J2)*10000)/10000
1550  LPRINT"X(";I;")=";INT(X(I, J2)*10000)/10000
1555  NEXT I
1560  FOR I=NO+1 TO N
1565  PRINT "L(";I-NO;")=";INT(X(I, J2)*10000)/10000
1570  LPRINT"L(";I-NO;")=";INT(X(I, J2)*10000)/10000
1575  NEXT I
1580  GOSUB 3500
1585  LPRINT "Object Function Jmin=";F
```

```
1590  PRINT
1595  PRINT "Object Function Jmin=";F
1600  END
2000  REM Calculate Object Function
2005  B=0
2010  FOR K=1 TO N0
2015  A=0
2020  FOR I=1 TO M0
2025  A=A+X((N0+I),J)*D(I,K)
2030  NEXT I
2035  B=B+(G(K)-A)*(G(K)-A)
2040  NEXT K
2045  C=0
2050  FOR I=1 TO M0
2055  C=C+W(I)^2
2060  NEXT I
2065  S(J)=B+C
2070  RETURN
3000  REM Object Function Express
3005  G(1)=4*X(1,J)-2*X(2,J)-6
3010  G(2)=-2*X(1,J)+4*X(2,J)
3015  G(3)=0
3020  G(4)=0
3025  W(1)=3*X(1,J)+4*X(2,J)+X(3,J)^2-6
3030  W(2)=-X(1,J)+4*X(2,J)+X(4,J)^2-2
3035  D(1,1)=3
3040  D(1,2)=4
3045  D(1,3)=2*X(3,J)
3050  D(1,4)=0
3055  D(2,1)=-1
3060  D(2,2)=4
3065  D(2,3)=0
3070  D(2,4)=2*X(4,J)
3075  GOSUB 2000
3080  RETURN
3500  F=2*X(1,J2)^2+2*X(2,J2)^2-X(1,J2)*X(2,J2)*2-6*X(1,J2)
3505  RETURN
```

程序执行前，要完成下列准备工作。

①目标函数表达式的列写：

例如目标函数是 J

$$J = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 6x_1$$

则在3500句开始编入子程序如下：

```
3500  F=2*X(1,J2)^2+2*X(2,J2)^2-2*X(1,J2)*X(2,J2)-6*X(1,J2)
3505  RETURN
```

②偏导数表达式及约束条件表达式的列写：

对于等式约束条件，由于没有引入松弛变量，因此，变量个数不发生变化，约束条件表达式可直接列写。例如，约束条件是：

$$3x_1 + 4x_2 - 6 = 0; -x_1 + 4x_2 - 2 = 0$$

写成BASIC语句表达式是：

```
W(1)=3*X(1,J)+4*X(2,J)-6
W(2)=-X(1,J)+4*X(2,J)-2
```

对于不等式约束条件，不等式约束条件方程个数是 m_0，则引入相应数目的松弛变量。总的变量个数是原有变量个数 n_0 与松弛变量个数 m_0 之和。例如不等式约束条件是：

$$3x_1 + 4x_2 - 6 \leqslant 0 \text{ 及 } -x_1 + 4x_2 - 2 \leqslant 0$$

则原变量数2个，引入 x_3 及 x_4 两个变量。

约束条件成为等式约束条件。要注意引入条件中的松弛变量应取平方。如上例中，可写成下列语句表达式：

```
W(1)=3*X(1,J)+4*X(2,J)-6+X(3,J)^2
W(2)=-X(1,J)+4*X(2,J)-2+X(4,J)^2
```

偏导数表达式包括两部分。用 $G(I)$ 表示目标函数对变量的偏导。这里的变量包括松弛变量。用 $D(I,K)$ 表示第 I 个约束条件对第 K 个变量的偏导。对上面所述的目标函数 J，对 x_1、x_2 的偏导是：

$$\frac{\partial J}{\partial x_1} = 4x_1 - 2x_2 - 6; \qquad \frac{\partial J}{\partial x_2} = 4x_2 - 2x_1$$

若为不等式约束条件，则有：

$$\frac{\partial J}{\partial x_3} = 0; \qquad \frac{\partial J}{\partial x_4} = 0$$

相应的BASIC语句可写成：

```
G(1)=4*X(1,J)-2*X(2,J)-6
G(2)=4*X(2,J)-2*X(1,J)
```

如为不等式约束条件，加入下列语句

```
G(3)=0
G(4)=0
```

约束条件的偏导也采用相同方法。即

$$\frac{\partial g_1(x)}{\partial x_1} = 3; \frac{\partial g_1(x)}{\partial x_2} = 4; \frac{\partial g_2(x)}{\partial x_1} = -1; \frac{\partial g_2(x)}{\partial x_2} = 4$$

不等式约束条件时，还包括对松弛变量的偏导。因此有：

$$\frac{\partial g_1(x)}{\partial x_3} = 2x_3; \frac{\partial g_1(x)}{\partial x_4} = 0; \frac{\partial g_2(x)}{\partial x_3} = 0; \frac{\partial g_2(x)}{\partial x_4} = 2x_4$$

在程序中，从3000语句开始到3070语句写入目标函数偏导，约束条件，约束条件偏导

的各语句，最后应加入3075语句及3080语句，当上述偏导表达式、约束条件表达式的语句较多时，按语句的顺序号，在最后加上GOSUB 2000及RETURN语句。这在使用时必须注意。2000语句开始的子程序是用于计算新的目标函数（6－49）的。

③初始值的输入

初始值按下列顺序送入，用DATA语句：

N0，M0，X（1，0），…，X（N0+M0，0）

共有N0+M0+2个数据。

N0：变量数。对于等式约束条件，它表示原变量个数。对于不等式约束条件，它还包括松弛变量个数，在输入时应予注意。

M0：约束条件个数。

而$X(I,0)$的初值包括了原变量、松弛变量及拉格朗日乘子的初值。

〔例6－5〕对具有不等式约束条件：

$$g_1(x) = 3x_1 + 4x_2 - 6 \leqslant 0$$

$$g_2(x) = -x_1 + 4x_2 - 2 \leqslant 0$$

目标函数

$$J = 2x_1^2 + 2x_2^2 - 2x_1x_2 - 6x_1$$

用拉格朗日乘子法计算最优解。

按程序清单给出的3500～3505语句输入目标函数表达式。用3000～3070语句输入目标函数偏导G(I)、约束条件W(I)及约束条件偏导D(I,K)的表达式。然后送入下列初始值数据语句：

5000 DATA 4，2，1，0，0.5，1，0，－0.5

则计算机经搜索后，在CRT与打印机同时输出下列结果。

X(1)=1.4594

X(2)=.4054

X(3)=－.0001

X(4)=1.3556

L(1)=－.3244

L(2)=－.0001

Object Function Jmin=－5.351351

应该说明，除了X(1)与X(2)是所需的最优解，J_{min}是极值外，X(3)与X(4)是松弛变量，L(1)与L(2)是拉格朗日乘子。此外，在程序段3000语句开始的子程序中，G(I)、W(I)及D(I,K)的先后次序是不影响程序执行结果的。

6.2.6 静态优化的实例

工业生产过程中，*PID*控制是最常见的控制规律，当被控对象已知时，如何选择合适的控制器参数，使控制系统的品质指标达到最优的问题是参数最优化问题。由于它与时间因素无关，因此属于静态优化问题。

代数方程$F(x)=0$的求解，通常可以采用方程求根的牛顿法、高斯法等求解。对于一些较为复杂的方程组，求解会变得十分复杂。如果采用目标函数$J=F^2(x)$则它的极小值必是方程的根。上面介绍的拉格朗日乘子法就是采用寻优的方法解决方程求根问题的。例如在频率特性的研究过程中，常常要求出$\angle G(j\omega) = -180°$的频率$\omega$，如果用目标函数$J=$

$(\angle G(j\omega)+180°)^2$，则最优解就是频率特性通过负实轴的频率。这类问题也属于静态优化问题。

在单纯形加速法中曾经介绍过用单纯形加速法对被控对象进行参数估计。这也属于参数优化的问题。在实际工业控制的应用中，可以举出很多静态优化的问题。下面是其中的两个实际例子。

6.2.6.1 控制器参数的最优选择

对于控制器参数的最优选择问题，其目标函数通常应选用能反应控制系统品质指标又能容易计算的。如前所述，通常选用积分型目标函数，例如 *ISE*、*ITAE* 等。对于同一个控制系统，采用不同的目标函数，所求得的最优控制器参数是不同的。所以在实际应用中常常发现同一被控对象，如反应器温度，在升温阶段（随动控制）与恒温阶段（定值控制）要采用不同的控制器参数，才能使在两个阶段的控制中都有较好的控制效果。当然在上述问题中，两个阶段中被控对象的特性也可能有变化。

控制器参数寻优的目标函数与前面讨论的不同之处是，通常目标函数不能用解析式来表达，它是通过控制系统的动态特性计算后获得。因此，对每一个搜索点都要计算控制系统的动态特性。在仿真程序上表现为寻优仿真程序与控制系统动态特性仿真程序是结合在一起的。

通常，不需要给出控制器参数寻优问题的约束条件。由于实际设备的限制，例如控制器输出不可能无限地增大或减小，控制阀的流能量不可能无限增大，因此，对控制器输出加入不等式约束条件，对有些问题（如差拍控制）就显得十分必要。有时，考虑到实际控制器参数不允许超出某一界限值，也可以加入相应的约束条件。应该指出，这些约束条件是在程序执行过程中用限幅等方式来实现的。

通常，对控制器参数寻优问题，采用单纯形加速法作为直接寻优的算法。其计算步骤是：

①给定对象参数，进行标幺化。

②给定控制器参数初值。

③调用单纯形加速法寻优程序。

④调用连续系统或离散系统数字仿真程序计算被控对象动态特性。

⑤计算目标函数值。

⑥判别目标函数是否达到极值，若不满足则转③，否则转⑦。

⑦打印控制器最优参数值。

有时，为了给出某类控制系统中控制器参数的最优整定规则，还要对不同组的对象参数分别求出相应的控制器参数的最优值，然后，进行统计、回归或者曲线拟合，得到该类控制系统的控制器参数整定规则公式或曲线。

〔例 6－6〕对由被控过程 $G(s)$ 与控制器 $G_c(s)$ 组成的单回路控制系统，研究比例、比例加积分、比例积分微分控制器最优参数与对象参数的关系。设

$$G(s)=\frac{Ke^{-\tau s}}{Ts+1}$$

对比例控制器，$G_c(s)=K_c$；对比例积分控制器，$G_c(s)=K_c\left(1+\frac{1}{T_i s}\right)$；对比例积分微分控制器：

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right)$$

采用 *ISE* 及 *ITAE* 作为目标函数，分别计算相应的最优参数值。通常把回路增益 KK_c、积分时间与对象时间常数之比$\frac{T_i}{T}$、微分时间与对象时间常数之比$\frac{T_d}{T}$与对象的纯滞后与对象时间常数之比$\frac{\tau}{T}$联系起来。即有

$$Y = A\left(\frac{\tau}{T}\right)^B \tag{6-50}$$

Y 按调节作用不同而分别是：

比例作用：$Y = KK_c$

积分作用：$Y = \frac{T}{T_i}$

微分作用：$Y = \frac{T_d}{T}$

洛佩兹（Lopez）给出了相应的 A 与 B 的数据（如表 6－1 所示），它适用于定值系统。

表 6－1　定值系统的整定参数值

目标函数	调节规律	调节作用	*A*	*B*
ISE	比　例	比　例	1.411	－0.917
ITAE	比　例	比　例	0.904	－1.084
ISE	比例积分	比　例	1.305	－0.959
		积　分	0.492	－0.739
ITAE	比例积分	比　例	0.859	－0.977
		积　分	0.674	－0.680
ISE	比例积分微分	比　例	1.495	－0.945
		积　分	1.101	－0.771
		微　分	0.560	1.006
ITAE	比例积分微分	比　例	1.357	－0.947
		积　分	0.842	－0.738
		微　分	0.381	0.995

洛佩兹还计算了随动系统的最优参数整定公式。其中，积分作用按下列公式：

$$Y = A + B\left(\frac{\tau}{T}\right) \tag{6-51}$$

其余部分的整定公式与（6－50）相同。表 6－2 是随动系统的最优整定参数值。

表 6－2　随动系统的整定参数值

目标函数	调节规律	调节作用	*A*	*B*
ITAE	比例积分	比　例	0.586	－0.916
		积　分	1.03	－0.165
ITAE	比例积分微分	比　例	0.965	－0.855
		积　分	0.796	－0.147
		微　分	0.308	0.929

6.2.6.2　控制系统稳定域

为了保证控制系统在稳定条件下运行，对组成控制器的参数有一个约束条件，这个约束

条件是使系统处于临界稳定状态时的参数值。对于纯比例控制的控制系统，它就是根轨迹与虚轴交点时的放大倍数。对于比例积分控制作用，用根轨迹法变得十分复杂。如果从频率特性来考虑，则可以从频率特性曲线与负实轴的交点来确定临界放大倍数。换句话说，就是求 $\angle G(j\omega) + 180^\circ = 0$ 的根 ω，并获得相应的 $K_c = \dfrac{1}{|G(j\omega)|}$。

用最优方法对方程求根，是选用一定的目标函数求极小。如方程为 $y = f(x) = 0$，则可选目标函数 $J = f^2(x)$，如图 6－2 所示，其最小值必为 0，相应的最优解必为方程的根。

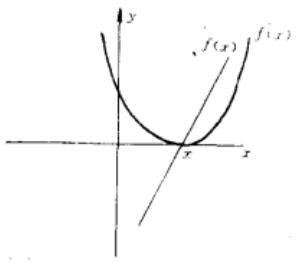


图 6－2　方程求根转化为求极小值

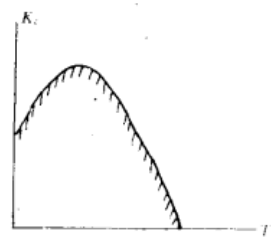


图 6－3　稳定域

下面以比例积分作用下的控制系统为例说明求取系统稳定域的计算步骤：

①T_i 置 0。

②设置 K_c 初值为 1，频率初值置某值。

③调用单纯形加速法调优程序。

④调用频率特性仿真程序。

⑤计算目标函数 $J = \underline{\angle G(j\omega)} + 180^\circ$ 的值。

⑥若满足最小值要求，转⑦，否则转③。

⑦计算 T_i 相应的 $K_c = \dfrac{1}{|G(j\omega)|}$，并打印 T_i 及 K_c。

⑧ $T_i \Leftarrow T_i + \Delta T_i$。

⑨判别终止条件是否满足，若满足则结束，否则转②。

图 6－3 是某控制系统，采用 *PI* 调节器时参数的稳定域。如果采用的 K_c 及 T_i 值在域内，系统是稳定的。在边界上，则是临界稳定的。在域外则控制系统不稳定。若采用衰减频率特性，则得到以该衰减比 $n:1$ 为边界的区域，同样，在域内的点，其衰减比大于 $n:1$。域外则衰减比小于 $n:1$。

6.3　动态最优化方法

6.3.1　变分法

6.3.1.1　变分法的基本概念

变分法是研究泛函极值的一种经典方法。泛函是函数概念的扩充。求泛函极值的方法与求函数极值的方法有许多类似之处。现把变分法的有关概念介绍如下：

①泛函：对于某一类函数集合中的每一个函数 $f(x)$，均有一个确定的数 J 与之对应，则称 J 为依赖于函数 $f(x)$ 的泛函。记为：

$$J = J[f(x)]$$

如同函数 $y(x)$ 规定了数 y 对应于数 x 一样，泛函规定了数 J 与函数 $f(x)$ 的对应关系，因此，泛函可认为是函数的函数。

③泛函的极值：如果泛函 $J[f(x)]$ 在任何一条与 $f=f_0(x)$ 接近的曲线上的值不小于 $J[f_0(x)]$，即

$$J[f(x)]-J[f_0(x)]\geqslant 0$$

则称泛函 $J[f(x)]$ 在曲线 $f_0(x)$ 上达到极小值。

在变分法中，有零阶接近度和一阶接近度两种意义的函数接近。

当函数 $f(x)$ 与 $f_0(x)$ 之差的绝对值，即

$$|f(x)-f_0(x)|$$

对定义域内一切 x 都很小时，则函数 $f(x)$ 与 $f_0(x)$ 有零阶接近度。

对定义域内一切 x，若满足上述条件外，还满足它们导数差的绝对值很小，即有

$$\begin{cases}|f(x)-f_0(x)|\leqslant\varepsilon\\ |\dot{f}(x)-\dot{f}_0(x)|\leqslant\varepsilon\end{cases}(x_1\leqslant x\leqslant x_2)$$

则称函数 $f(x)$ 与 $f_0(x)$ 有一阶接近度。

定义变分 $\delta f(x)$ 为两函数之差，即

$$\delta f(x)\triangleq f(x)-f_0(x)$$

②泛函的变分：泛函 $J[f(x)]$ 的增量可表示为：

$$\begin{aligned}\Delta J&=J[f(x)+\delta f(x)]-J[f(x)]\\&=\delta J+r[f(x),\delta f(x)]\end{aligned}$$

式中 δJ 称为泛函的变分。$r[f(x),\delta f(x)]$ 是高阶无穷小。

函数的微分是函数增量的线性主部，同样，泛函的变分是泛函增量的线性主部。

〔例 6-7〕 求 $J=\int_{t_0}^{t_1}F[t,x(t),\dot{x}(t)]dt$ 的变分

$$\begin{aligned}\delta J&=\frac{\partial}{\partial\alpha}J[f+\alpha\delta f]\big|_{\alpha=0}\\&=\int_{t_0}^{t_1}\frac{\partial}{\partial\alpha}F[t,x(t)+\alpha\delta x(t),\dot{x}(t)+\alpha\delta\dot{x}(t)]\big|_{\alpha=0}dt\\&=\int_{t_0}^{t_1}\left\{\frac{\partial F[t,x(t),\dot{x}(t)]}{\partial x(t)}\cdot\partial x(t)+\frac{\partial F[t,x(t),\dot{x}(t)]}{\partial\dot{x}(t)}\partial\dot{x}(t)\right\}dt\end{aligned}$$

④欧拉方程：前面已经指出，泛函极值存在的必要条件是泛函的一次变分为 0，即

$$\delta J=0$$

设被控过程是：

$$\dot{x}(t)=\varphi[x(t),u(t,t]\tag{6-52}$$

状态初值：$x(t_0)=x_0$

状态终止时刻是 t_f，$t_f>t_0$，终端状态满足

$$g[x(t_f),t_f]=0\tag{6-53}$$

试用变分的方法求取容许控制中的一组最优控制 $u^*(t)$，使系统从 x_0 转移到终端 $x(t_f)$ 时，使目标函数 J

$$J=\int_{t_0}^{t_f}F[x(t),u(t),t]dt\tag{6-54}$$

为极小值（或极大值）。

根据被控过程的状态方程，可求得 $u(t)$ 的表达式，代入（6－54），可得

$$J = \int_{t_0}^{t_f} F[x(t), \dot{x}(t), t]dt \tag{6-55}$$

设 $x^*(t)$ 是最优轨线，用

$$x(t) = x^*(t) + a\delta x(t) \tag{6-56}$$

表示最优轨线邻域内的容许曲线。$\delta x(t)$ 是 $x(t)$ 的变分，它应连续可导，并满足：

$$\delta x(t_0) = \delta x(t_f) = 0$$

同样，应用变分的导数等于导数的变分，得：

$$\dot{x}(t) = \dot{x}^*(t) + a\delta\dot{x}(t) \tag{6-57}$$

代入目标函数式，得：

$$J(a) = \int_{t_0}^{t_f} F[x^*(t) + a\delta x(t), \dot{x}^*(t) + a\delta\dot{x}(t), t]dt$$

根据泛函极值存在的必要条件，有

$$\left.\frac{\partial J(a)}{\partial a}\right|_{a=0} = 0 \tag{6-58}$$

$F(x, \dot{x}, t)$ 有二阶连续偏导数，故求导和积分次序可以互换，从而有：

$$\left.\frac{\partial J(a)}{\partial a}\right|_{a=0} = \int_{t_0}^{t_f}\left\{\frac{\partial F[x^*(t), \dot{x}^*(t), t]}{\partial x(t)}\delta x(t) + \frac{\partial F[x^*(t), \dot{x}^*(t), t]}{\partial \dot{x}(t)}\delta\dot{x}(t)\right\}dt = 0$$

对第二项进行分部积分，可得：

$$\int_{t_0}^{t_f}\left[\frac{\partial F}{\partial x} - \frac{d}{dt}\frac{\partial F}{\partial \dot{x}}\right]\delta x dt = 0 \tag{6-59}$$

根据变分学基本引理，可得：

$$\frac{\partial F}{\partial x} - \frac{d}{dt}\frac{\partial F}{\partial \dot{x}} = 0 \tag{6-60}$$

上式称为欧拉—拉格朗日方程。

上面的讨论没有考虑终端的约束条件，在约束条件下，可以推导出同样的方程式。只是横截条件不同。欧拉方程是二阶常微分方程，其积分曲线 $x(t) = x(t, c_1, c_2)$ 含有两个待定常数，可以用边界条件确定。

〔例 6－8〕 已知 $x(0) = 0, x\left(\frac{\pi}{2}\right) = 1$，求目标函数的最优轨线 $x(t)$。设目标函数

$$J = \int_0^{\frac{\pi}{2}}(\dot{x}^2 - x^2)dt$$

$\because$ $$F = \dot{x}^2 - x^2$$

欧拉方程： $\ddot{x} + x = 0$

解： $x(t) = c_1\cos t + c_2\sin t$

代入边界条件，有 $c_1 = 0$，$c_2 = 1$，故最优轨线是：

$$x(t) = \sin t$$

6.3.1.2 等式约束条件下的变分问题

对于给定的泛函

$$J = \int_{t_0}^{t_f} F[x(t), \dot{x}(t), t]dt$$

在等式约束条件

$$\psi[x(t), \dot{x}(t), t] = 0 \tag{6-61}$$

下的极值问题，可用引入拉格朗日乘子的方法，引入哈密尔顿函数（Hamilton）

$$H[x(t), \dot{x}(t), \lambda(t), t] = F[x(t), \dot{x}(t), t] + \lambda^T \psi[x(t), \dot{x}(t), t] \tag{6-62}$$

使泛函

$$J = \int_{t_0}^{t_f} H[x(t), \dot{x}(t), \lambda(t), t]dt$$

达到无条件极值。因此有：

$$\frac{\partial H}{\partial x} - \frac{d}{dt}\frac{\partial H}{\partial \dot{x}} = 0 \tag{6-63}$$

最优解 $x(t)$ 与 $\lambda(t)$ 由（6-63）与约束条件方程（6-61）求解得出。

〔例 6-9〕已知状态方程：

$$\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

要求初态 $\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$，终态 $\begin{bmatrix} x_1(2) \\ x_2(2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ 时，目标函数 $J = \frac{1}{2}\int_0^2 u^2(t)dt$ 最小的控制 $u(t)$。

令 $H = \frac{1}{2}u^2(t) + \lambda^T(t)[Ax(t) + Bu(t) - \dot{x}(t)]$

∴泛函成为

$$J = \int_0^2 Hdt = \int_0^2 \left\{ \frac{1}{2}u^2(t) + \lambda_1(t)[x_2(t) - \dot{x}_1(t)] + \lambda_2(t)[u(t) - \dot{x}_2(t)] \right\} dt$$

解下列方程：

$$\begin{cases} \dot{\lambda}_1(t) = 0 \\ \dot{\lambda}_2(t) = -\lambda_1(t) \\ u(t) = -\lambda_2(t) \\ \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = u(t) \end{cases}$$

得通解：$\lambda_1(t) = c_1; \lambda_2(t) = -c_1 t + c_2; u(t) = c_1 t - c_2;$

$$x_1(t) = \frac{1}{6}c_1 t^3 - \frac{1}{2}c_2 t^2 + c_3 t + c_4; x_2(t) = \frac{1}{2}c_1 t^2 - c_2 t + c_3$$

代入边界条件，得：$c_1 = 3; c_2 = \frac{7}{2}; c_3 = 1; c_4 = 1$

因此，最优控制：$u(t) = 3t - \frac{7}{2}$，相应的状态轨线是：$x_1(t) = \frac{1}{2}t^3 - \frac{7}{4}t^2 + t + 1;$

$x_2(t) = \frac{3}{2}t^2 - \frac{7}{2}t + 1$。

6.3.1.3 有终端约束的变分问题

设被控过程是：

$$\dot{x}(t) = \psi[x(t), u(t), t] \tag{6-64}$$

初始状态是

$$x(t_0) = x_0 \tag{6-65}$$

终止状态 $t = t_f$ 时满足：

$$g[x(t_f), t_f] = 0 \tag{6-66}$$

寻找最优控制 $u(t)$，使泛函 J

$$J = S[x(t_f), t_f] + \int_{t_0}^{t_f} F[x(t), u(t), t]dt \tag{6-67}$$

为极值。

引入拉格朗日乘子 λ，并构造哈密尔顿函数 H：

$$H = F[x, u, t] + \lambda^T \psi[x, u, t] \tag{6-68}$$

引入待定乘子向量 μ，构造新的终值目标函数：

$$\theta[x(t_f), t_f] = S[x(t_f), t_f] + \mu^T g[x(t_f), t_f] \tag{6-69}$$

则新的泛函成为 J

$$J = \theta[x(t_f), t_f] + \int_{t_0}^{t_f} \{H[x, u, \lambda, t] - \lambda^T \dot{x}\}dt \tag{6-70}$$

而乘子 λ 与状态 $x(t)$ 应满足条件：

①规范方程：$-\dfrac{\partial H}{\partial x} = \dot{\lambda}(t)$ (6-71)

即：

$$\dot{\lambda}(t) = -\frac{\partial F[x, u, t]}{\partial x} - \frac{\partial \psi[x, u, t]}{\partial x}\lambda \tag{6-72}$$

②状态方程：$\dot{x}(t) = \psi[x, u, t]$ (6-73)

根据边界条件有：

③ $x(t_0) = x_0$ (6-74)

④ $\lambda(t_f) = \dfrac{\partial S[x(t_f), t_f]}{\partial x(t_f)} + \mu \dfrac{\partial g[x(t_f), t_f]}{\partial x(t_f)}$ (6-75)

⑤ $g[x(t_f), t_f] = 0$ (6-76)

根据极值条件有：

⑥ $\dfrac{\partial H}{\partial \hat{u}} = 0$ (6-77)

解上述方程，即可得最优控制轨线 $u^*(t)$。

〔例 6-10〕 状态方程 $\begin{pmatrix} \dot{x}^1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$

初态 $\begin{pmatrix} x_1(0) \\ x_2(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$，$t_f = 1$，终止约束条件是：

$$x_1(t_f) + x_2(t_f) - 1 = 0$$

求 $J = \dfrac{1}{2}\int_0^1 u^2 dt$ 极值时的最优控制 $u^*(t)$。

列出下列方程：

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x_1} = 0$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial \dot{x}_2} = -\lambda_1$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = u$$

$$x_1(0) = 0$$

$$x_2(0) = 0$$

$$\lambda_1(1) = \frac{\partial g}{\partial x_1}\mu = \mu$$

$$\lambda_2(1) = \frac{\partial g}{\partial x_2}\mu = \mu$$

$$x_1(1) + x_2(1) = 1$$

$$\frac{\partial H}{\partial u} = u + \lambda_2 = 0$$

式中：$H = \frac{1}{2}u^2 + \lambda_1 x_2 + \lambda_2 u$

由此解得最优控制轨线：$u^*(t) = -\frac{3}{7}t + \frac{6}{7}$，相应的状态轨线是：$x_1(t) = -\frac{1}{14}t^3 + \frac{3}{7}t^2$；$x_2(t) = -\frac{3}{14}t^2 + \frac{6}{7}t$

对于终止时刻 t_f 未定的情况可仿上述方法处理。即把 t_f 也作为控制变量处理。除了上述的方程（6－72）～（6－77）外，还应加入哈密尔顿函数在最优轨线终端满足：

$$H[x^*(t_f^*), u^*(t_f^*), \lambda(t_f^*), t_f^*] = -\frac{\partial \theta[x^*(t_f^*), t_f^*]}{\partial t_f^*} \tag{6-78}$$

上述条件是终端 t_f 无约束的结果。其余情况可类似处理。

根据上述，采用变分法求解泛函极值问题，可以通过引入哈密尔顿函数，转化为代数方程求根问题。

6.3.2 极大值原理

古典变分法的应用条件过严，使有些寻优问题无法解决。它的局限性表现为：

①对控制量 $u(t)$ 不加限制；而实际情况下，控制量的大小是受约束的。对这类约束条件，古典变分法无法求解。

②对函数 F、ψ、S 等要求可微，尤其是哈密尔顿函数 H，应存在 $\frac{\partial H}{\partial u}$，否则将无法求导，从而不能用古典变分法求解。

对此，不少学者进行了研究，庞特里雅金（Понтрягин）的极大值原理（又称最小值原理）和贝尔曼（Bellman）的动态规划较为成功。取得了广泛的应用。

6.3.2.1 极大值原理

下面以终值型目标函数 $J = S[x(t_f)]$，对定常系统：$\dot{x}(t) = \psi(x, u)$，$x(t_0) = x_0$，终值 t_f 未知时，求取最优控制轨线 $u^*(t)$ 为例说明极大值原理。

设哈密尔顿函数：

$$H(x, u, \lambda) = \lambda^T(t)\psi(x, u) \tag{6-79}$$

设 $u^*(t)$ 和 t_f^* 是使 J 最小的最优解。则必存在不为零的向量 $\lambda(t)$，使 $\lambda(t)$ 是方程

$$\dot{\lambda}(t) = -\frac{\partial H(x, u, \lambda)}{\partial x} \tag{6-80}$$

满足边界条件

$$\lambda(t_f) = \frac{\partial S[x(t_f)]}{\partial x(t_f)} \tag{6-81}$$

的解。则 $u^*(t), x^*(t), t_f^*$ 满足：

$$H[x^*(t), u^*(t), \lambda(t)] = \min_{u(t)\in U} H[x^*(t), u(t), \lambda(t)] \tag{6-82}$$

或

$$H[x^*(t), u^*(t), \lambda(t)] \leqslant H[x^*(t), u(t), \lambda(t)] \text{ 对 } t \in [t_0, t_f], u(t) \in U$$

及当 t_f 自由时，有

$$H[x^*(t), u^*(t), \lambda(t)] = H[x^*(t_f^*), u^*(t_f^*), \lambda(t_f^*)] = 0 \tag{6-83}$$

当 t_f 固定时，有

$$H[x^*(t), u^*(t), \lambda(t)] = H[x^*(t_f), u^*(t_f), \lambda(t_f)] = \text{常数} \tag{6-84}$$

可以看到，极大值原理与古典变分法的不同处在于：

①控制变量允许的范围有了约束，即有：

$$u(t) \in U \tag{6-85}$$

②极值条件发生变化。古典变分法中，有

$$\frac{\partial H}{\partial u} = 0$$

而极大值原理中，则有（6－82）式成立。这使得最优控制可能取 U 集的边界上的值成为可能。因此变分法是极大值原理的特例。

由于庞氏在证明过程中，取 λ 与 H 与上述符号相反，因此，其结果是最优解在 H 函数取极大值处。因此称为极大值原理。这里沿用这种习惯叫法，但其实质是最小值。

对极大值原理的证明，这里不作介绍，有兴趣的读者可参考有关文献。下面介绍极大值原理的几种具体形式。

(1) 非定常系统

设非定常系统 $\dot{x}(t) = \psi[x(t), u(t), t]$，初值 $x(t_0) = x_0$，终止时间 t_f 未知，$t \in [t_o, t_f]$，求最优控制 $u(t)$，它满足约束条件

$$u(t) \in U$$

并使终值型目标函数 J

$$J = S[x(t_f), t_f]$$

为最小。

定义哈密尔顿函数 $H[x(t), u(t), \lambda(t), t]$

$$H[x(t), u(t), \lambda(t), t] = \lambda^T(t)\psi[x(t), u(t), t]$$

列出下列方程：

①协态方程：$\dot{\lambda}(t) = -\dfrac{\partial H[x(t), u(t), \lambda(t), t]}{\partial x(t)}$ (6－86)

②状态方程：$\dot{x}(t) = \dfrac{\partial H[x(t), u(t), \lambda(t), t]}{\partial \lambda} = \psi[x(t), u(t), t]$ (6－87)

③初始条件：$x(t_0) = x_0$ (6－88)

④横截条件：$\lambda(t_f) = \dfrac{\partial S[x(t_f), t_f]}{\partial x(t_f)}$ (6－89)

求出哈密尔顿函数极小值时的 $u^*(t)$，使

⑤ $$H[x^*(t),u^*(t),\lambda(t),t]=\min_{u(t)\in U}H[x^*(t),u(t),\lambda(t),t] \tag{6-90}$$

成立。

由于 t_f 未知，则 t_f^* 应满足：

⑥ $$H[x^*(t_f^*),u^*(t_f^*),\lambda(t_f^*),t_f^*]=-\frac{\partial S[x^*(t_f^*),t_f^*]}{\partial t_f} \tag{6-91}$$

最优轨线上的哈密尔顿函数满足

⑦ $$H[x(t),u(t),\lambda(t),t]=H[x(t_f),u(t_f),\lambda(t_f),t_f]+\int_{t_f}^{t}\frac{\partial H[x(s),u(s),\lambda(s),s]}{\partial s}ds \tag{6-92}$$

(2) 积分型目标函数

对上述问题，如目标函数改为积分型指标：

$$J=\int_{t_0}^{t_f}F[x(t),u(t),t]dt \tag{6-93}$$

则除了列出①、②、③三个方程外，由于终值目标 $S[x(t_f),t_f]=0$，故方程④成为

④横截条件：$\lambda(t_f)=0$

即自然边界条件。

根据极大值原理，方程⑤同样应满足。而最优轨线的终端哈密尔顿函数可根据 t_f 的情况分为：

⑥当 t_f 未固定时，应满足：

$$H[x^*(t),u^*(t),\lambda(t)]=H[x^*(t_f^*),u^*(t_f^*),\lambda(t_f^*)]=0 \tag{6-94}$$

当 t_f 固定时，应满足：

$$H[x^*(t),u^*(t),\lambda(t)]=H[x^*(t_f^*),u^*(t_f^*),\lambda(t_f^*)]=\text{常数} \tag{6-95}$$

6.3.2.2 极大值原理求解最优控制问题

用极大值原理求解最优控制问题的一般计算步骤如下：

①列写哈密尔顿函数 H：

$$H[x(t),u(t),\lambda(t),t]=F[x(t),u(t),t]+\lambda^T\psi[x(t),u(t),t]$$

②求解 $u^*(x,\lambda,t)$，使哈密尔顿函数取极小值。

当$\frac{\partial H}{\partial u}$存在时，由

$$\frac{\partial H}{\partial u}=0$$

求出 $u^*(x,\lambda,t)$；

当$\frac{\partial H}{\partial u}$不存在时，由

$$H[x^*(t),u^*(t),\lambda(t),t]=\min_{u(t)\in U}H[x(t),u(t),\lambda(t),t]$$

求出 $u^*(x,\lambda,t)$。

③求解协态方程、状态方程组及 $u^*(x,\lambda,t)$。即有

$$\dot{x}=\psi[x(t),u(t),t]$$

$$\dot{\lambda}=-\frac{\partial H}{\partial x}$$

用 $u^*(x,\lambda,t)$ 代入上述方程组求出 $x(t),\lambda(t)$。

④用初始条件、横截条件解两点边值问题。

初始条件为 $x(t_0)=x_0$，而终止条件 $\lambda(t_f)$ 则根据不同的边界条件确定。两点边值问题可用解析法或梯度法等求解。结果为最优状态轨线 $x^*(t)$ 与最优协状态轨线 $\lambda^*(t)$。

⑤代入 $u^*(x,\lambda,t)$ 得最优控制轨线 $u^*(t)$。

⑥用二次变分或对目标函数进行一些计算来验证极值的充分条件是否满足。

〔例 6－11〕已知系统状态方程：

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix}=\begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}u(t)$$

初态 $x(0)=\begin{bmatrix}x_{01}\\ x_{02}\end{bmatrix}$；求最优控制 $u^*(t)$，使满足

$$|u(t)|\leqslant 1$$

且使 $x(t_f)=\begin{bmatrix}0\\ 0\end{bmatrix}$ 时有最短时间。其求解过程如下：

对最短时间控制，有 $F=1$，因此 $J=\int_{t_0}^{t_f}dt=t_f-t_0$

①列出哈密尔顿函数 H：

$$H=1+\lambda^TAx+\lambda^TBu=1+\lambda_1(t)x_2(t)+\lambda_2u(t)$$

②协态方程求解：

$$\because\quad \dot{\lambda}(t)=-\frac{\partial H}{\partial x}$$

$$\therefore\quad \begin{bmatrix}\dot{\lambda}_1(t)\\ \dot{\lambda}_2(t)\end{bmatrix}=-\begin{bmatrix}\dfrac{\partial H}{\partial x_1}\\ \dfrac{\partial H}{\partial x_2}\end{bmatrix}=\begin{bmatrix}0\\ -\lambda_1(t)\end{bmatrix}$$

初态 $\lambda_1(0)=\lambda_{01};\lambda_2(0)=\lambda_{02}$，$\therefore\lambda_1(t)=\lambda_{01};\lambda_2(t)=\lambda_{02}-\lambda_{01}t$

③按极大值原理确定控制序列 $u^*(x,\lambda,t)$：

$$1+\lambda_1(t)x_2^*(t)+\lambda_2u^*(t)\leqslant 1+\lambda_1(t)x_2(t)+\lambda_2(t)u(t)$$

$$\therefore\quad u^*(t)=-\operatorname{sgn}[\lambda_2(t)]=-\operatorname{sgn}[\lambda_{02}-\lambda_{01}t]$$

根据 λ_{01}、λ_{02}的取值，有四种可能的 $u^*(t)$ 取值。

④计算 $x(t)$：

根据 $\dot{x}(t)=Ax(t)+Bu(t)$ 得

$$x(t)=e^{At}x(t_0)+\int_{t_0}^{t}e^{A(t-\tau)}Bu(\tau)d\tau$$

$$=\begin{bmatrix}x_{01}+x_{02}t\\ x_{02}\end{bmatrix}+\begin{bmatrix}\frac{1}{2}t^2\\ t\end{bmatrix}u(t)$$

根据 $|u(t)|\leqslant 1$，故 $u(t)$ 取值为 ± 1，

解得：$\quad x_1(t)=x_{01}-\frac{1}{2}x_{02}^2u(t)+\frac{1}{2}x_2^2(t)u(t)$

当 $u(t)=1$ 时，有：$x_1(t)=\left(x_{01}-\frac{1}{2}x_{02}^2\right)+\frac{1}{2}x_2^2(t)$

当 $u(t)=-1$ 时，有：$x_1(t)=\left(x_{01}+\frac{1}{2}x_{02}^2\right)-\frac{1}{2}x_2^2(t)$

图 6－4 是状态轨线图。

⑤确定最优控制 $u^*(t)$。

最优控制 $u^*(t)$ 与初始状态 $\begin{bmatrix}x_{01}\\x_{02}\end{bmatrix}$ 有关，根据最短时间控制的要求，$u(t)$ 只有 ±1 的取值，四种可选的序列。

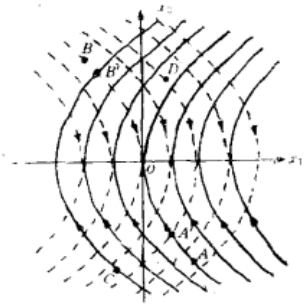


图 6－4　状态轨线图

· $u^*(t)=\{1\}$：当 $\begin{bmatrix}x_{01}\\x_{02}\end{bmatrix}$ 在图 6－4 所示轨线的 AO 段上任一点时，最优控制 $u^*(t)=1$。

· $u^*(t)=\{-1\}$：当 $\begin{bmatrix}x_{01}\\x_{02}\end{bmatrix}$ 在图示轨线的 BO 段上任一点时，最优控制 $u^*(t)=-1$。

· $u^*(t)=\{+1,-1\}$：当 $\begin{bmatrix}x_{01}\\x_{02}\end{bmatrix}$ 在轨线 AOB 的左侧时，如图中的 C 点，则控制作用 $u(t)$ 先为 1，到达 B' 点（在 BO 轨线上）时切换为 $u(t)=-1$，然后到达终态。

· $u^*(t)=\{-1,+1\}$：当 $\begin{bmatrix}x_{01}\\x_{02}\end{bmatrix}$ 在轨线 AOB 的右侧时，如图中的 D 点，则控制作用 $u(t)$ 先是 -1，到达 A' 点（在 AO 轨线上）时切换为 $u(t)=1$，然后到达终态。

可以看到，开关的切换点必在 AOB 轨线上。此外，当状态初值确定后，最优控制作用是唯一确定的。

6.3.2.3　线性二次型最优控制问题

对于线性被控对象，若状态方程为：

$$\dot{x}(t)=Ax(t)+Bu(t);x(t_0)=x_0$$

采用线性二次型的综合目标函数

$$J=\frac{1}{2}x^T(t_f)Fx(t_f)+\frac{1}{2}\int_{t_0}^{t_f}[x^T(t)Qx(t)+u^T(t)Ru(t)]dt \qquad (6-96)$$

当 t_0、t_f 及 $x(t_0)$ 已知时，求最优控制 $u^*(t)$，使目标函数 J 为极小的问题称为线性二次型问题，又称线性状态调节器问题。

采用极大值原理进行求解的步骤如下：

①列出哈密尔顿函数 H：

$$H[x,u,\lambda,t]=\frac{1}{2}x^TQx+\frac{1}{2}u^TRu+\lambda^TAx+\lambda^TBu \qquad (6-97)$$

②列出哈密尔顿方程（即状态方程和协态方程）：

$$\frac{\partial H}{\partial x}=-\dot{\lambda}(t)=Qx+A^T\lambda \qquad (6-98)$$

$$\frac{\partial H}{\partial \lambda}=Ax+Bu=\dot{x} \qquad (6-99)$$

③极值条件可由 $\frac{\partial H}{\partial u}=0$ 求得：

$$\frac{\partial H}{\partial u}=Ru+B^T\lambda=0 \qquad (6-100)$$

④终端条件由下式确定：

$$\lambda(t_f) = \left.\frac{\partial\left[\frac{1}{2}x^TFx\right]}{\partial x}\right|_{t=t_f} = Fx(t_f) \qquad (6-101)$$

⑤为使 $u(t)$ 是状态 $x(t)$ 的反馈形式，令

$$\lambda(t) = P(t)x(t)$$

根据（6－100）得：

$$u(t) = -R^{-1}B^T\lambda(t) = -R^{-1}B^TP(t)x(t) \qquad (6-102)$$

⑥黎卡提（Riccati）方程的列写：

由（6－102）代入（6－99），根据

$$\dot{\lambda}(t) = \dot{P}(t)x(t) + P(t)\dot{x}(t)$$
$$\dot{x}(t) = Ax(t) - BR^{-1}B^TP(t)x(t)$$
$$\dot{\lambda}(t) = -Qx - A^TP(t)x(t)$$

整理得黎卡提方程：

$$-\dot{P}(t) = P(t)A + A^TP(t) - P(t)BR^{-1}B^TP(t) + Q \qquad (6-103)$$

其终值条件是 $P(t_f) = F$。

⑦求解黎卡提方程，得 $P(t)$，则最优控制 $u^*(t)$

$$u^*(t) = -L(t)x(t) \qquad (6-104)$$

式中，$L(t) = R^{-1}B^TP(t)$ (6－105)

可以看到，最优反馈控制矩阵 L 的计算，取决于黎卡提方程的求解。$P(t)$ 的稳态解可以通过求解代数黎卡提方程

$$PA + A^TP - PBR^{-1}B^TP + Q = 0$$

求得。为便于计算机计算，现介绍克莱曼（Kleinmann）的迭代法。迭代法的计算步骤如下：

①$k \Leftarrow 0$，$P(k) \Leftarrow 0$。

②计算 $G = BR^{-1}B^T$。

③$A(k) = A - GP(k)$。

④解李雅普诺夫方程

$$A^T(k)P(k+1) + P(k+1)A(k) + P(k)GP(k) + Q = 0 \qquad (6-106)$$

得 $P(k+1)$。

⑤判别 $P(k+1)$ 是否收敛到稳态，如满足则转⑧，否则转⑥。

⑥$k \Leftarrow k+1$。

⑦判别迭代次数是否到，若满足，则询问是否增加迭代次数，不增加则结束，否则转④。

⑧打印结果，并结束计算。

对于李雅普诺夫方程的求解，采用了加倍递推迭代算法，可见有关文献的介绍。

线性二次型最优控制规则的计算程序如下，程序中的变量说明如下：

N：状态变量个数。

M：控制变量个数。

E：精度。

T：迭代次数。

A1(I,J)：状态方程中矩阵 A。

B(I,J)：状态方程中矩阵 B。

Q1(I,J)：状态权矩阵 Q。

Q2(I,J)：控制权矩阵 R。

S0(I,J)：状态反馈矩阵中的 P。

L(I,J)：状态反馈矩阵。

本程序运行前的准备工作是送入下列数据：

N, M, A1(I,J), B(I,J), Q1(I,J), Q2(I,J), S0(I,J)

例如，送入下列语句

```
5000  DATA 2,1,0,1,-1,0,0,1,1,0,0,100,1,5,5,5,5
```

打印结果如下：

```
A(1,1)=0
A(1,2)=1
A(2,1)=-1
A(2,2)=0
B(1,1)=0
B(2,1)=1
Q(1,1)=1
Q(1,2)=0
Q(2,1)=0
Q(2,2)=100
R(1,1)=1
.....
L(1,1)=.4142083
L(1,2)=10.04134
```

计算结果是针对下列问题的。

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$J = \int_0^\infty \left(x^T \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix} x + u^T \cdot 1 \cdot u \right) dt$$

最优控制：

$$u^*(t) = -\begin{bmatrix} 0.414 & 10.04 \end{bmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = -0.414x_1(t) - 10.04x_2(t)$$

对于不同的权函数，结果不同。这表明可以通过权函数的选择，来改变最优控制轨线。下面是 Q 取不同值时的计算结果。

```
Q(1,1)=100
Q(1,2)=0
Q(2,1)=0
Q(2,2)=10
```

R(1,1)=1

.....

L(1,1)=9.049871

L(1,2)=5.300918

求线性二次型最优控制律的仿真程序如下：

```
100  REM Calculate Coefficients of Optimal Feedback Controller
105  CLS
110  LOCATE 2,1
115  READ N:REM N:The Number of The System States
120  LOCATE 4,1
125  READ M:REM M:The Number of The Control Variables
130  DIM A(N,N),A1(N,N),B(N,M),U(M,N),F1(N,N),F2(N,N),F3(N,N)
135  DIM S0(N,N),Q(N,N),F0(N,N),R0(N,N),Q1(N,N),Q2(M,M)
140  GOSUB 3500
145  E=.0001:REM Accuracy E
150  T=20:REM Iterational Number
155  REM PRINT "Accuracy of Calculation E=";E
160  REM PRINT "Iterational Nnmber T=";T
165  Q=0
170  Q1=0
175  Q0=0
180  GOSUB 1000
185  FOR I=1 TO N
190  FOR J=1 TO M
195  F1(I,J)=0
200  FOR K=1 TO M
205  F1(I,J)=F1(I,J)+B(I,K)*U(K,J)
210  NEXT K
215  NEXT J
220  FOR J=1 TO N
225  R0(I,J)=0
230  FOR K=1 TO M
235  R0(I,J)=R0(I,J)+F1(I,K)*B(J,K)
240  NEXT K
245  NEXT J
250  NEXT I
255  IF Q1<>1 THEN 275
260  INPUT "Input Additional Iterational Number Q0=";Q0
265  Q1=0
270  T=T+Q0
```

```
275  FOR I=1 TO N
280  FOR J=1 TO N
285  F1(I,J)=0
290  F0(I,J)=0
295  FOR K=1 TO N
300  F1(I,J)=F1(I,J)+R0(I,K)*S0(K,J)
305  F0(I,J)=F0(I,J)+S0(I,K)*R0(K,J)
310  NEXT K
315  NEXT J
320  FOR J=1 TO N
325  Q(I,J)=0
330  FOR K=1 TO N
335  Q(I,J)=Q(I,J)+F0(I,K)*S0(K,J)
340  NEXT K
345  NEXT J
350  NEXT I
355  FOR I=1 TO N
360  FOR J=1 TO N
365  A(I,J)=A1(I,J)-F1(I,J)
370  Q(I,J)=Q1(I,J)+Q(I,J)
375  NEXT J
380  NEXT I
385  Q=Q+1
390  CLS:LOCATE 10,1:PRINT"The Iterational Number is ";Q
395  GOSUB 1500
400  C0=0
405  C1=0
410  FOR J=1 TO N
415  C2=0
420  C3=0
425  FOR I=1 TO N
430  C2=C2+ABS(F2(I,J)-S0(I,J))
435  C3=C3+ABS(S0(I,J))
440  NEXT I
445  IF C0>C2 THEN 455
450  C0=C2
455  IF C1>C3 THEN 465
460  C1=C3
465  NEXT J
470  IF C1=0 THEN 480
```

```
475  C=C0/C1
480  IF C-E<0 THEN 550
485  IF Q-T<0 THEN 520
490  PRINT "Accuracy Unsatisfied After";T;"Times Iterational Operation"
495  PRINT "If Continue Then Type Y else N";
500  INPUT" ";A$
505  Q1=0
510  IF A$="Y" OR A$="Y" THEN Q1=1
515  IF Q1=0 THEN 550
520  FOR I=1 TO N
525  FOR J=1 TO N
530  S0(I,J)=F2(I,J)
535  NEXT J
540  NEXT I
545  GOTO 255
550  FOR I=1 TO M
555  FOR J=1 TO N
560  F1(I,J)=0
565  FOR K=1 TO M
570  F1(I,J)=F1(I,J)+U(I,K)*B(J,K)
575  NEXT K
580  NEXT J
585  FOR J=1 TO N
590  F3(I,J)=0
595  FOR K=1 TO N
600  F3(I,J)=F3(I,J)+F1(I,K)*F2(K,J)
605  NEXT K
610  NEXT J
615  FOR J=1 TO N
620  F3(I,J)=0
625  FOR K=1 TO N
630  F3(I,J)=F3(I,J)+F1(I,K)*F2(K,J)
635  NEXT K
640  NEXT J
645  NEXT I
650  GOSUB 3000
655  END
1000  REM Inverse Matrix
1005  FOR I=1 TO M
1010  FOR J=1 TO M
```

```
1015  U(I, J) = 0
1020  NEXT J
1025  U(I, I) = 1
1030  NEXT I
1035  FOR J = 1 TO M
1040  C0 = 0
1045  C1 = 0
1050  FOR I = J TO M
1055  IF ABS(Q2(I, J)) < = C1 THEN 1070
1060  C0 = I
1065  C1 = ABS(Q2(I, J))
1070  NEXT I
1075  IF C1 < >0 THEN 1090
1080  PRINT "The matrix Q2 can not be inversed"
1085  GOTO 625
1090  I = C0
1095  FOR K = 1 TO M
1100  C = Q2(J, K)
1105  Q2(J, K) = Q2(I, K)
1110  Q2(I, K) = C
1115  C = U(J, K)
1120  U(J, K) = U(I, K)
1125  U(I, K) = C
1130  NEXT K
1135  C1 = 1/Q2(J, J)
1140  FOR K = 1 TO M
1145  Q2(J, K) = C1 * Q2(J, K)
1150  U(J, K) = C1 * U(J, K)
1155  NEXT K
1160  FOR I = 1 TO M
1165  IF I = J THEN 1195
1170  C1 = - Q2(I, J)
1175  FOR K = 1 TO M
1180  Q2(I, K) = Q2(I, K) + C1 * Q2(J, K)
1185  U(I, K) = U(I, K) + C1 * U(J, K)
1190  NEXT K
1195  NEXT I
1200  NEXT J
1205  RETURN
1500  REM F2 = S(T) + ···
```

```
1505  U=.000001
1510  S=0
1515  FOR J=1 TO N
1520  S1=0
1525  FOR I=1 TO N
1530  S1=S1+ABS(A(I,J))
1535  NEXT I
1540  IF S1<S THEN 1550
1545  S=S1
1550  NEXT J
1555  S=1/S
1560  GOSUB 2000
1565  GOSUB 2500
1570  N0=1
1575  FOR I=1 TO N
1580  FOR J=1 TO N
1585  F1(I,J)=0
1590  F0(I,J)=0
1595  F3(I,J)=0
1600  FOR K=1 TO N
1605  F1(I,J)=F1(I,J)+F(K,I)*F2(K,J)
1610  F3(I,J)=F3(I,J)+F(I,K)*F(K,J)
1615  NEXT K
1620  NEXT J
1625  FOR J=1 TO N
1630  FOR K=1 TO N
1635  F0(I,J)=F0(I,J)+F1(I,K)*F(K,J)
1640  NEXT K
1645  NEXT J
1650  NEXT I
1655  FOR I=1 TO N
1660  FOR J=1 TO N
1665  F(I,J)=F3(I,J)
1670  F2(I,J)=F2(I,J)+F0(I,J)
1675  IF N0=1 THEN F0(I,J)=F2(I,J)
1680  NEXT J
1685  NEXT I
1690  H$="True"
1695  N0=N0+1
1700  FOR I=1 TO N
```

```
1705  FOR J=1 TO N
1710  IF ABS(F0(I,J))<ABS(U*F2(I,J)) THEN 1730
1715  H$="False"
1720  J=N
1725  I=N
1730  NEXT J
1735  NEXT I
1740  IF H$="True"OR N0=20 THEN 1750
1745  GOTO 1575
1750  RETURN
2000  REM F=Exp(AT)
2005  FOR I=1 TO N
2010  FOR J=1 TO N
2015  F(I,J)=0
2020  F2(I,J)=0
2025  F(I,I)=1
2030  F2((I,I)=1
2035  N0=0
2040  NEXT J
2045  NEXT I
2050  FOR I=1 TO N
2055  FOR J=1 TO N
2060  F1(I,J)=0
2065  F3(I,J)=0
2070  FOR K=1 TO N
2075  F1(I,J)=F1(I,J)+F2(I,K)*A(K,J)*S/(N0+1)
2080  NEXT K
2085  F(I,J)=F(I,J)+F1(I,J)
2090  NEXT J
2095  NEXT I
2100  FOR I=1 TO N
2105  FOR J=1 TO N
2110  F2(I,J)=F1(I,J)
2115  NEXT J
2120  NEXT I
2125  FOR I=1 TO N
2130  FOR J=1 TO N
2135  FOR K=1 TO N
2140  F3(I,J)=F3(I,J)+F2(I,K)*A(K,J)*S/(N0+2)
2145  NEXT K
```

```
2150  NEXT J
2155  NEXT I
2160  H$ = "True"
2165  N0 = N0 + 1
2170  FOR I = 1 TO N
2175  FOR J = 1 TO N
2180  IF ABS(F3(I, J)) < ABS(U * F(I, J)) THEN 2200
2185  H$ = "False"
2190  J = N
2195  I = N
2200  NEXT J
2205  NEXT I
2210  IF H$ = "True" THEN 2225
2215  IF N0 = 50 THEN 2225
2220  GOTO 2050
2225  RETURN
2500  REM F2
2505  FOR I = 1 TO N
2510  FOR J = 1 TO N
2515  F2(I, J) = S * Q(I, J)
2520  F3(I, J) = S * Q(I, J)
2525  N0 = 0
2530  NEXT J
2535  NEXT I
2540  FOR I = 1 TO N
2545  FOR J = 1 TO N
2550  F1(I, J) = 0
2555  FOR K = 1 TO N
2560  F1(I, J) = F1(I, J) + (A(K, I) * F3(K, J) + F3(I, K) * A(K, J)) * S/(N0 + 2)
2565  NEXT K
2570  NEXT J
2575  NEXT I
2580  FOR I = 1 TO N
2585  FOR J = 1 TO N
2590  F3(I, J) = F1(I, J)
2595  F2(I, J) = F2(I, J) + F1(I, J)
2600  NEXT J
2605  NEXT I
2610  H$ = "True"
2615  N0 = N0 + 1
```

```
2620  FOR I=1 TO N
2625  FOR J=1 TO N
2630  IF ABS(F3(I,J))<ABS(U*F2(I,J)) THEN 2650
2635  H$="False"
2640  J=N
2645  I=N
2650  NEXT J
2655  NEXT I
2660  IF H$="True" OR N0=100 THEN 2670
2665  GOTO 2540
2670  RETURN
3000  GOSUB 4000:REM Print
3005  LPRINT"·····":LPRINT
3010  FOR I=1 TO M
3015  FOR J=1 TO N
3020  LPRINT"L(";I;",";J;")=";F3(I,J)
3025  NEXT J
3030  NEXT I
3035  RETURN
3500  REM Input State Matrix
3505  FOR I=1 TO N
3510  FOR J=1 TO N
3515  READ A1(I,J):REM State Matrix
3520  NEXT J
3525  NEXT I
3530  REM Control Matrix
3535  FOR I=1 TO N
3540  FOR J=1 TO M
3545  READ B(I,J):REM Control Matrix
3550  NEXT J
3555  NEXT I
3560  REM Weighting Matrix Q
3565  FOR I=1 TO N
3570  FOR J=1 TO N
3575  READ Q1(I,J):REM Wighting Matrix Q
3580  NEXT J
3585  NEXT I
3590  REM Wighting Matrix R
3595  FOR I=1 TO M
3600  FOR J=1 TO M
```

```
3605  READ Q2(I,J):REM Wighting matrix R
3610  NEXT J
3615  NEXT I
3620  REM Matrix S0
3625  FOR I=1 TO N
3630  FOR J=1 TO N
3635  READ S0(I,J):REM Matrix S0
3640  NEXT J
3645  NEXT I
3650  RETURN
4000  LPRINT:REM Lprint Matrix A
4005  FOR I=1 TO N
4010  FOR J=1 TO N
4015  LPRINT "A(";I;",";J;")=";A1(I,J):REM State Matrix
4020  NEXT J
4025  NEXT I
4030  LPRINT:REM Lprint Matrix B
4035  FOR I=1 TO N
4040  FOR J=1 TO M
4045  LPRINT "B(";I;",";J;")=";B(I,J):REM Control Matrix
4050  NEXT J
4055  NEXT I
4060  LPRINT:REM Lprint Weighting Matrix Q
4065  FOR I=1 TO N
4070  FOR J=1 TO N
4075  LPRINT "Q(";I;",";J;")=";Q1(I,J):REM Wighting Matrix Q
4080  NEXT J
4085  NEXT I
4090  LPRINT:REM Lprint Wighting Matrix R
4095  FOR I=1 TO M
4100  FOR J=1 TO M
4105  LPRINT "R(";I;",";J;")=";Q2(I,J):REM Wighting matrix R
4110  NEXT J
4115  NEXT I
4120  LPRINT
4125  RETURN
```

〔例 6－12〕已知被控过程的状态方程为：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ -4 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}u$$

求使目标函数

$$J = \frac{1}{2}\int_0^{\infty}\left\{x^T\begin{bmatrix}400 & 200\\ 200 & 100\end{bmatrix}x + u^T[1.667]u\right\}dt$$

最优的控制。

调用本程序，并输入下列语句：

5000 DATA 2,1,0,1,-4,-2,0,1,400,200,200,100,1.667,0,0,0,0

经迭代运行后，打印出下列反馈系数：

L (1, 1) =11.9985

L (1, 2) =7.38003

因此，最优控制 $u(t)$ 由下式给出。

$$u(t) = -[11.9985 \quad 7.38]\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} = -11.9985x_1(t) - 7.38x_2(t)$$

6.3.3 动态规划法

动态规划法是1957年贝尔曼（Bellman）等学者提出的多阶段决策方法。这种方法的基本内容是把给定的原始问题分解成若干阶段或者若干子问题，对每一个子问题进行决策，然后逐级推进，最后一级所得的最优解就是原始问题的最优解。由于各阶段采取的决策与时间有关，故称为动态规划。

6.3.3.1 多级决策过程

下面以简单的例子说明多级决策过程。

如图6-5所示的最短路线问题。要求从 A 地到 E 地找出最短的路线。各地间的距离如图示。

下面介绍几个符号及其意义：

n：级变量。它表示从起点到某点的级数。如图中，$B(B_1, B_2)$ 是第一级或第一阶段。

x_n：状态变量。它表示在 n 级所处的位置。

$u_n(x)$：决策变量。

$d_n(x_n, u_n)$：效益。

用实例说明上述概念。

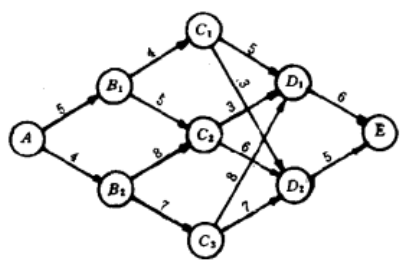


图 6-5 最短路线问题

〔例6-13〕用动态规划法解下列线性规划问题。

已知约束条件

$$4u_1 + 2u_2 \leqslant 12; u_1 \geqslant 0; u_2 \geqslant 0$$

求目标函数

$$J = 8u + 10u_2$$

的极大值。

把问题分解为两阶段。这里，每一阶段只有一个变量。设第一阶段只考虑变量 u_1，则有阶段效益：$d_1(x_1, u_1) = 8u_1$。

在第二阶段，只有变量 u_2，效益为：

$$d_2(x_2, u_2) = 10u_2$$

总的效益

$$J = d(x_1, u_1) + d(x_2, u_2) = 8u_1 + 10u_2$$

对约束条件同样进行分解。对每一阶段，只需满足约束条件的一部分，在最后一级才完全满足约束条件。

对第一阶段，u_1 的取值范围由

$$0 \leqslant 4u_1 \leqslant 12$$

决定。状态变量 $x_2 = 12 - 4u_1 \geqslant 0$

对第二阶段，变量 u_2 满足：$x_3 = 12 - 4u_1 - 2u_2 \geqslant 0$。由于 $x_2 = 12 - 4u_1$。所以，有

$$x_3 = x_2 - 2u_2 \geqslant 0$$

状态初值用 x_1 表示，$x_1 = 12$。

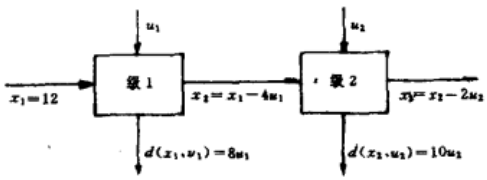


图 6－6　阶段的分解

因此，可以列出下列图形表示状态、决策及效益的关系。

动态规划法是从后阶段向前阶段推进，逐级求出最优策略。下面予以说明。

①对第二阶段，效益函数 $f_2(x_2)$ 为：

$$f_2(x_2) = \max_{u_2 \in U_2} \{d_2(x_2, u_2)\} = \max_{u_2 \in U_2} \{10u_2\}$$

式中，$U_2 = \{u_2 \mid x_2 - 2u_2 \geqslant 0,\ u_2 \geqslant 0\}$

$$\therefore \quad f_2(x_2) = \max_{0 \leqslant u_2 \leqslant 0.5x_2} \{10u_2\}$$

$$u_2^* = 0.5x_2, \therefore f_2^*(x_2) = 5x_2$$

②对第一阶段，效益函数 $f_1(x_1)$ 为：

$$f_1(x_1) = \max_{u_1 \in U_1} \{d_1(x_1, u_1) + f_2(x_2)\}$$

$$= \max_{u_1 \in U_1} \{8u_1 + 5x_2\}$$

式中，$U_1 = \{u_1 \mid 4u_1 \leqslant x_1, u_1 \geqslant 0\}$

$$\because \quad x_2 = x_1 - 4u_1$$

$$\therefore \quad f_1(x_1) = \max_{0 \leqslant u_1 \leqslant 0.25x_1} \{8u_1 + 5(x_1 - 4u_1)\} = \max_{0 \leqslant u_1 \leqslant 0.25x_1} \{5x_1 - 12u_1\}$$

$$u_1^* = 0, \therefore f_1^*(x_1) = 5x_1$$

$$\because \quad x_1 = 12, \therefore f^*(x_1) = 5x_1 = 60。$$

相应求得：$x_2 = x_1 - 4u_1 = 12$；$u_1^* = 0$；$u_2^* = 0.5x_2 = 6$。

6.3.3.2　最优化原则

根据动态规划的决策确定过程可以看到，一个过程的最优策略具有这样的性质，即无论其初始状态和初始决策如何，其今后的所有决策，对第一个决策所形成的状态作为初始状态的过程（即局部多级过程）而言，必须构成最优策略，这就是动态规划的最优化原则。因此，可以得到最优化原则的递推公式：

$$f_n(x) = \min_{u_n(x)} \{d[x, u_n(x)] + f_{n+1}[u_n(x)]\} \tag{6-107}$$

$$f_{N+1}(x) = d(x, E) \tag{6-108}$$

式中 N 为总的阶段数。

动态规划法从最末阶段开始，逐级求出前一阶段的控制变量，使最末阶段达到最优的目标函数。最后算出由初始状态的条件下的各阶段最优的控制变量和状态变量。由此可见，动态规划的计算包括两次搜索。第一次是逆向进行的，从指定的终态计算 $f_{N+1}(x)$，然后计

算 $f_N(x),\cdots$。第二次是状态的正向搜索，它从第一次搜索所得的决策 $u^*(x^*(k))$ 出发，按状态方程求出决策序列和最优轨线。

当目标函数有良好解析性质时，每一阶段的求极值可用求导或用差商代替求导在计算机上求出。而没有良好解析性质的目标函数的情况，常要占用大量内存单元存放效益函数的值。

动态规划的计算步骤如下：

①计算 $f_{N+1}(x^{(j)}), j = 1,2,\cdots,J$，结果存高速存储器。

②$k = N$。

③$j = 1$。

④$m = 1$。

⑤计算 x^{j+1}。

⑥若 x^{j+1} 未整量化，则进行插值计算。

⑦计算 $f_{k+1}(x) + d[x, u_k(x)]$。

⑧若 $m = M$ 则转⑨，否则 $m = m + 1$，转⑤。

⑨对 $f_{k+1}(x) + d[x, u_k(x)]$ 求极小值 $f_k(x)$，并得相应的最优控制 $u_k^*(x)$。

⑩$f_{k+1}(x)$ 存入低速存储器。

⑪若 $j = J$ 则转⑫，否则 $j = j + 1$，转④。

⑫若 $k = 0$ 则结束，否则 $f_k(x)$ 存高速存储器，$k = k - 1$，转③。

动态规划法适用于有约束条件的线性或非线性系统的寻优，对目标函数和约束条件是否有解析式无要求，且可以获得全局最优解。但对 x 与 u 维数的增加会成倍增加存储量和计算时间。

7 离散事件系统的数字仿真

前面各章所讨论的系统，其输入变量都假定是确定性的。而在实际系统中，由于存在扰动和量测噪声，因此，系统带有不确定性。这种系统称为随机系统。对于确定性系统，我们可以用解析函数的方法来描述系统的输入、状态和输出之间的关系。对于随机系统，由于扰动的特征是不可能精确地预报其未来值，因此，不能用解析函数进行描述，而必须用统计的概念来作扰动的模型。一个比较成功的方法就是用随机变量序列作为扰动的模型。维纳（Wiener）和郭尔莫哥洛夫（Kolmogorov）开创的滤波和预报理论是随机控制理论的里程碑。约瑟夫（Joseph）和图（Tou）发表的分离定理提供了滤波理论和最优随机控制理论之间的联系。线性二次高斯系统，即用二次性能指标的带高斯分布随机干扰的线性系统是分离定理的应用。由于篇幅限制，本章对这些随机控制系统的数字仿真虽未涉及，但是对于随机过程中的基本概念、随机数产生和变换等都作了较详细的介绍，这些知识对于随机控制系统的数字仿真是十分有用的。本章所讨论的离散事件系统数字仿真是针对系统的状态只在离散时刻发生变化，即用离散事件来表征这些随机变化的系统，并据此分析和研究这类系统的。

7.1 基本概念

7.1.1 随机变量

在随机试验中，样本空间中的每一个样本点都有一个实值函数的变量相对应，这个变量称为随机变量，用 $X(\omega)$ 表示。随机变量的取值不能在试验前得知，它取决于试验的结果。因此，随机变量是试验结果的函数。

随机变量用概率分布函数来描述。把 $X(\omega)$ 取值不大于 x 作为事件 A，即

$$A = \{X(\omega) \leqslant x\}$$

则事件 A 的概率 $P(A)$

$$P(A) = P\{X(\omega) \leqslant x\}$$

称为随机变量 $X(\omega)$ 在 x 处的概率分布函数，简称分布函数，用 $F(x)$ 表示。

$$F(x) = P(A) = P\{X(\omega) \leqslant x\} \tag{7-1}$$

概率密度函数 $f(x)$，简称密度函数，定义为：

$$f(x) = \lim_{\Delta x \to 0} \frac{F(x + \Delta x) - F(x)}{\Delta x} = \frac{dF(x)}{dx} \tag{7-2}$$

概率分布函数具有下列性质：

① $F(x)$ 是单调递增函数。即随着 x 的增大，出现 $X(\omega)$ 大于 x 的事件概率也越大。

② $F(-\infty) = 0; F(+\infty) = 1; 0 \leqslant F(x) \leqslant 1$。

③ $F(x)$ 处处左连续。

同样，概率密度函数具有下列性质：

① $f(x) \geqslant 0$ 是非负函数。

② $\int_{-\infty}^{+\infty} f(x)dx = 1$。

③ $F(x) = \int_{-\infty}^{x} f(t)dt$。

〔例 7－1〕正态分布

正态分布是最常见也是最重要的一个随机变量的分布函数。它的密度函数是：

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}} \tag{7-3}$$

正态分布密度函数的图形曲线是对称于 $x = a$ 的，在 $x = a$ 处有极大值 $\frac{1}{\sqrt{2\pi}\sigma}$，在 $x = a \pm \sigma$ 处曲线有拐点，当 $x \to \pm\infty$ 时，曲线以 x 轴为渐近线。

改变参数 a 的值，分布密度曲线沿 x 轴平移，其形状不发生变化。改变参数 σ 的值，引起极大值的变化。同时，为使曲线面积保持为 1，随 σ 的减小，曲线中心部分纵坐标值增大，两侧很快趋于 x 轴。

服从正态分布的随机变量 X 在区间 $(a - k\sigma, a + k\sigma)$ 内的概率见表 7－1。

表 7－1　正态分布时的概率

k	1	2	3	4	5
$P(\lvert X-a\rvert < k\sigma)$	0.6826	0.9544	0.9973	0.99994	0.9999994

随机变量的函数是随机变量。随机变量 X 的函数 Y 定义为：当随机变量 X 取值 x 时，随机变量 Y 取值 $y = \varphi(x)$。记为 $Y = \varphi(X)$。

〔例 7－2〕求线性函数 $y = kx + b(k \neq 0)$ 的分布密度。

根据 $y = kx + b$，有

$$x = g(y) = \frac{y-b}{k}; g'(y) = \frac{1}{k}。$$

$$\therefore \quad \varphi(y) = \frac{1}{|k|}\varphi\left(\frac{y-b}{k}\right) \tag{7-4}$$

若 X 服从正态分布，即

$$\varphi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}}$$

则 $y = kx + b$ 的分布密度函数

$$\varphi(y) = \frac{1}{2\pi|k|\sigma} e^{-\frac{(y-ka-b)^2}{2k^2\sigma^2}} \tag{7-5}$$

即服从正态分布的随机变量 x，它的线性函数 $y = kx + b$ 也服从正态分布。

在实际应用中，许多情况下，我们不需要知道随机变量的全部分布特征；有些情况下，则难于获得全部特征，而应用时，也只须了解随机变量的某些特征数字就够了，这些特征数字可由分布函数确定，或由分布函数经运算求得。随机变量的数字特征中最重要的是数学期望、方差和各阶矩。

数学期望是随机变量所有可能值的以其相应概率或概率密度为权的加权平均数。即

$$a = E[x] = \int_{-\infty}^{+\infty} xf(x)dx \qquad \text{（连续型）} \tag{7-6}$$

$$a = E[x] = \sum_{i=1}^{n} x_i p_i \qquad \text{（离散型）} \tag{7-7}$$

方差 V 是标准差 σ 的平方，即 σ^2，它是衡量随机变量 X 与它的均值 a 间分散程度的量。而标准差 σ 是 $X-a$ 的平方的数学期望。即

$$\sigma = \sqrt{E[(x-a)^2]} \tag{7-8}$$

随机变量 X 分布的关于 $x=x_0$ 的 k 阶距定义为：

$$E[(X-x_0)^k] = \int_{-\infty}^{\infty}(x-x_0)^k f(x)dx \tag{7-9}$$

当 x_0 是原点时，得到原点矩。当 $x_0 = E[X]$ 时，得到中心矩。

〔例 7-3〕正态分布随机变量的中心矩。

由定义得：

$$V_k = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}(x-a)^k e^{-\frac{(x-a)^2}{2\sigma^2}}dx$$

取置换积分变量 $t=\frac{x-a}{\sigma}$，得 k 阶中心矩

$$V_k = \frac{\sigma^k}{\sqrt{2\pi}}\int_{-\infty}^{\infty}t^k e^{-\frac{t^2}{2}}dt$$

①k 为奇数，被积函数是奇函数，有

$$V_k = 0 \qquad (k = 1,3,5\cdots) \tag{7-10}$$

②k 为偶数，取 $t^2=2z$，得：

$$V_k = (k-1)!!\sigma^k \qquad (k = 2,4,6,\cdots) \tag{7-11}$$

式中，$(k-1)!!$ 表示从 1 到 $(k-1)$ 的一切奇数之积。例如，正态分布的 4 阶中心矩

$$V_4 = 3\sigma^4$$

7.1.2 随机过程

随机变量 X 是样本空间 Ω 中样本点 ω 的函数 $X(\omega)$。随机过程 $\{x(t), t \in T\}$ 则是两个自变量的函数：$\{x(t,\omega), t \in T, \omega \in \Omega\}$。对于确定的 t，$t \in T$，$x(t,\cdot)$ 是随机变量。对于确定的 $\omega, \omega \in \Omega$，$x(\cdot,\omega)$ 是时间函数，它是随机过程的一个实现，或一个样本函数。因此，随机过程是取决于时间 t 的随机向量集。

用联合分布函数 $F_{t_1t_2\cdots t_n}(x_1,x_2,\cdots,x_n)$ 来描述随机过程 $\{x(t,\omega), t \in T, \omega \in \Omega\}$ 的统计特征。

随机过程比随机变量多一个时间的自变量。因此，对随机变量数字特征的定义也适用于随机过程。

随机过程分布函数等统计特征不随时间平移而变化时，称为平稳随机过程。若随机过程在 $\lim\limits_{\tau\to\infty}V(\tau) = 0$ 条件下，$T\to\infty$ 时的时间均值 $a_x(T)$ 与总体均值 $a_x(t)$ 相等，则称随机过程具有均值的遍历性，亦称各态历经性。

典型的随机过程有正态过程、马尔柯夫过程和白噪声过程。随机过程的联合分布函数族是正态分布的，称该过程是正态过程，或称高斯过程。实际应用中，随机噪声可认为是许多独立的正态分布的各随机噪声之和。若随机过程对 k 个时间点 $t_1<t_2<\cdots<t_k$，在任一点 t_i 瞬时的条件概率仅取决于其上一时刻 t_{i-1} 的值而与其以前的其他时刻的值无关，则称该过程为马尔柯夫过程。平稳随机过程的自相关函数是理想的脉冲函数时，称为白噪声过程。白噪声过程的功率谱密度是常数。

7.2 随机数的产生和变换

7.2.1 伪随机数的产生

7.2.1.1 对伪随机数的要求

用计算机按数学公式或者按移位寄存器的移位操作产生的随机数不是真正的随机数，为与真正的随机数相区别，称为伪随机数。数字仿真中所使用的随机数是伪随机数，下面为方便起见，简称为随机数。产生伪随机数的方法有多种，评价产生伪随机数的好坏有一些基本要求。主要是：

①要求产生的伪随机数序列有较理想的随机性和均匀性。

②产生伪随机数的计算机程序应简短、运算速度快、占用内存少。

③伪随机数序列的循环周期应尽可能长。

④能满足一定的精确度要求。

⑤伪随机数序列中，前后之间，各子序列之间要求相互独立。

7.2.1.2 同余数法

同余数法是近来在计算机上产生伪随机数序列最常用的一种方法。

对于模 M，数 x 与 y 同余是指 $(x-y)$ 能为 M 整除，用下式表示：

$$x \equiv y \pmod{M}$$

通常按一定的迭代关系由同余数法产生伪随机数。

(1) 加同余数法

常用的加同余数法采用下列迭代算式：

$$x_{i+2} \equiv x_i + x_{i+1} \pmod{M} \tag{7-12}$$

当 $x_0=1$，$x_1=1$ 时，如 $M=100$ 则可产生伪随机数序列为 1，1，2，3，5，8，13，21，34，55，89，44，33，77，10，87，97，84，81，65，……。

当 $x_0=1$，$x_1=1$ 时，如 $M=2^{44}\approx 1.75\times 10^3$ 时，伪随机数序列的循环周期是 $3\times 2^{43}\approx 2.5\times 10^3$。即最大周期 $T=1.5\times M$。由于它的统计特性较差，较少应用。

(2) 乘同余数法

乘同余数法的迭代算式是：

$$x_{i+1} \equiv a x_i \pmod{M} \tag{7-13}$$

当选择 $M=2^\beta$ 时，可以证明，最大循环周期发生在 $a=8r\pm 3$ 处。此时，r 是任意一正整数。通常取 β 为 31，48 或 35。例如，取 $\beta=31$，$a=5^{13}$，有

$$x_{i+1} \equiv 5^{13} x_i \pmod{2^{31}}$$

计算机的仿真计算步骤如下：

①选择初始种子值 x_0，$0<x_0<M=2^{31}$。

②选择乘数 a，取 $a=5^{13}=1220703125$。

③计算 ax_i，结果送 x_i。

④判别 x_i 是否大于 M，不大于时转⑥。

⑤x_i-M 送 x_i 单元，转④。

⑥计算 $u_i=x_i/M$，得到 0～1 之间的伪随机数。

⑦判别产生序列是否结束，如未结束则转③，否则子程序退出。

应该指出，在采用移位操作时，④⑤两步也可以用 x_i 存放单元的最低 β 位作为新的 x_i。

(3) 混合同余数法

混合同余数法的迭代算式是：

$$x_{i+1} = ax_i + c \qquad (\text{mod } M) \tag{7-14}$$

可以证明，由本算法得到的伪随机数序列的循环周期达到最大值 M 的条件是：

①c 与 M 无公约数。

②$a-1$ 是 M 的每一个素因子 g 的倍数。

③如果 M 是 4 的倍数，则 $a-1$ 是 4 的倍数。

满足上述条件并有较好统计特性的参数是：

$$M = 2^{35}; a = 2^7 + 1 = 129; C = 1。$$

7.2.1.3 伪随机数发生器的联合使用

由于单独用一种伪随机数发生器产生的伪随机数序列有较明显的规律性，统计特性较差。因此，在应用时可作下列改进。

(1) 等间隔跳跃选用和轮换使用

在所需随机数不多，而伪随机数发生器的循环周期很长时，可以等间隔地选用一个，把其余的舍去。这可用一句循环语句实现。

在一个仿真程序中，用几个随机数发生器（可不同类型，也可同一类型不同参数）轮换交替产生伪随机数，组合起来使用。

(2) 同余数法和移位寄存器法联合使用

移位寄存器法移位操作产生伪随机数的基本思想是：对一个 2^n 位的数 x_i，先右移 b_1 位，进行不进位的按位加，得到中间结果 w_i，把结果 w_i 再左移 b_2 位，再进行按位加，其结果就是 x_{i+1}。

一种采用乘同余数法和移位寄存器法联合使用的方法是先用乘同余数法产生一个伪随机数，然后用移位寄存器法由该数得到一个伪随机数。

产生的伪随机数可通过统计检验鉴定其随机性和均匀性。常用的方法是频率检验，又称 χ^2 检验及升降检验等。

7.2.2 随机变量的产生

随机变量产生的基本来源是 (0,1) 间的均匀分布随机数序列。有若干种算法可用于从已给的分布中产生随机变量。选用算法时应考虑下列诸因素：

①算法的精确度。

②算法所需的存贮容量及执行时间。

③算法本身的复杂程度。

7.2.2.1 逆变换法

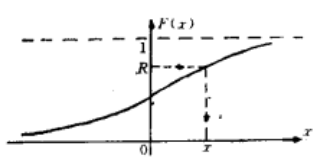


图 7-1 逆变换法的几何解释

该方法是基于下列定理。

定理：如果随机变量 X 的分布函数 $F(x)$ 是连续函数，则 $y = F(x)$ 是服从 (0, 1) 均匀分布的新的随机变量。

逆变换法利用上述定理，首先产生一个 (0, 1) 均匀分布的随机数 R，根据所需的分布 $F(x)$，确定其反函数 $x =$

$F^{-1}(y)$。然后，令 $x = F^{-1}(R)$ 得到服从分布为 $F(x)$ 的随机变量 X。图 7－1 是逆变换法的几何解释。可以看到，由于 $0<R<1$，而 $F(x)$ 定义在（0，1）内，因此，由 $F^{-1}(R)$ 即可求得 x。

逆变换法根据逆变换公式 $x = F^{-1}(R)$ 可直接得到随机变量的值，方法简便，但有些分布的逆变换公式不易用解析式表达时就无法采用本方法。下面介绍几个常用分布的例子。

（1）负指数分布

在研究排队、排序等离散事件系统中，相邻两个顾客或工件的到达间隔时间服从负指数分布，其分布密度

$$f(x) = \lambda e^{-\lambda x} \qquad (x \geqslant 0)$$

式中，$1/\lambda$ 是平均间隔时间。求得分布函数

$$F(x) = \int_0^x f(t)dt = 1 - e^{-\lambda x} \quad x \geqslant 0$$

其逆变换是：

$$x = F^{-1}(R) = -\frac{1}{\lambda}\ln(1 - R)$$

由于 R 是（0，1）上的均匀分布随机数，故（$1-R$）也是（0，1）上均匀分布随机数，因此服从负指数分布的随机数可用下列计算公式：

$$x = -\ln R/\lambda \tag{7-15}$$

（2）几何公布

在产品的抽样检验中，伯努利试验是指：若产品合格率为 p，则废品率为 $1-p=q$，若抽样时前 x 件均为废品，$x+1$ 件是合格品的概率可用几何分布描述。即

$$f(x) = pq^x$$

求得分布函数

$$F(x) = \sum_{i=0}^{x} pq^i = 1 - q^{x+1}$$

因此有逆变换公式：

$$q^{x+1} = 1 - R$$

为方便计算可写成：

$$q^x = R$$

即

$$x = \ln R/\ln g \tag{7-16}$$

（3）均匀分布

在 (a,b) 间均匀分布 的随机数的分布密度是：

$$f(x) = \begin{cases} \dfrac{1}{b-a} & a \leqslant x \leqslant b \\ 0 & \text{其余} \end{cases}$$

因此，有分布函数

$$F(x) = \begin{cases} 0 & x < a \\ \dfrac{x-a}{b-a} & a \leqslant x \leqslant b \\ 1 & x > b \end{cases}$$

故逆变换公式是：

$$x = F^{-1}(R) = a + (b - a)R \tag{7-17}$$

7.2.2.2 舍选法

舍选法适用于密度函数 $f(x)$ 在有限区间上有上界的场合。其方法是：

①产生两个随机数 R_1 和 R_2。

②根据 R_1 得到密度函数 $f(x)$ 所在区间〔a，b〕上均匀分布随机数 x

$$x = a + (b - a)R_1$$

并根据所给 $f(x)$ 求出 $f[a + (b - a)R_1]$ 的值。

③判别 $R_2 \leqslant f[a + (b - a)R_1]/M$ 是否成立。若成立则所得 x 为随机变量的值。若不成立，则舍弃，并回到①重新计算。式中 M 是 $f(x)$ 在〔a，b〕内的最大值。

舍选法的几何意义如图 7－2 所示。在边长为 $b-a$ 及 M 的矩形内，随机取点 p，若 p 位于 $f(x)$ 曲线上或曲线下部，则 p 的横坐标 x 就作为随机变量的值，如 p_1 点。否则就舍弃，如 p_2 点。

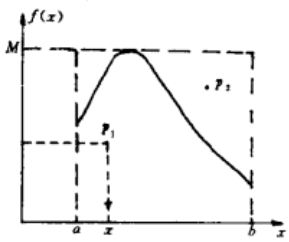


图 7－2　舍选法几何意义

下面介绍用舍选法产生随机变量值的例子。

(1) Beta 分布

在离散事件系统仿真中，对一些统筹问题常遇到某些活动或操作的持续时间，这些随机变量服从 Beta 分布。其分布密度函数

$$f(x) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} x^{r-1}(1-x)^{s-1} \qquad 0 \leqslant x \leqslant 1 \tag{7-18}$$

式中，$\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}dx$。

Beta 分布的最大值 M 可由 $\frac{df(x)}{dx} = 0$ 求得，即：

$$x = \frac{r-1}{s+r-2}$$

时有极大值 M。

参数 r 和 s 可根据所需 Beta 分布的均值和方差由下式计算：

$$r = E(x)\left[\frac{E(x)[1-E(x)]}{V(x)} - 1\right] \tag{7-19}$$

$$s = [1 - E(x)]\frac{r}{E(x)} \tag{7-20}$$

式中，$E(x)$、$V(x)$ 分别是 Beta 分布的均值和方差。由于 Beta 分布定义在（0，1）区间，因此，采用舍选法时 $a=0$，$b=1$。

(2) 正态分布

正态分布在离散事件系统的噪声仿真时经常采用。其密度函数是：

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-a)^2}{2\sigma^2}}$$

当均值 $a=0$，均方差 $\sigma=1$ 时称为标准正态分布。正态分布用 $N(a,\sigma^2)$ 表示。

藉助于有关定理，用舍选法得到标准正态分布的计算步骤如下：

①产生伪随机数 R_1、R_2。

②计算 $V_1=2R_1-1$；$V_2=2R_2-1$；$S=V_1^2+V_2^2$。由于 R_1、R_2 是（0，1）均匀分布

的随机数，显然，V_1、V_2 是（-1，1）均匀分布随机数。

③判别 $S \leqslant 1$。若满足转④，否则转①。

④计算 $A = \sqrt{-2\frac{\ln S}{S}}$；$X = V_1 A$；$Y = V_2 A$。则 X，Y 即为 $N(0,1)$ 的随机数。

7.2.2.3 组合法

若随机变量 X 的分布函数 $F(x)$ 不易用逆变换法表示时，可把 $F(x)$ 分解为若干 $F_i(x)$ 组合而成。每段 $F_i(x)$ 的概率为 p_i，即有：

$$F(x) = \sum_{i=1}^{\infty} p_i F_i(x)$$

而对于每一区段的分布 $F_i(x)$，可以据此求得随机变量的值。则组合法的计算步骤是：

①产生随机数 R；

②根据随机数 R，确定区段，并由此决定按该段分布 $F_i(x)$ 得到随机变量的值。

（1）泊松分布（Poisson）

如前述，两个事件出现的间隔时间服从负指数分布，而单位时间内，事件出现的次数服从泊松分布。其分布密度是：

$$f(x) = \frac{\lambda^x}{x!} e^{-\lambda} \qquad x = 0,1,2,\cdots \tag{7-21}$$

由于第 i 间隔的时间 $y_i = -\frac{1}{\lambda}\ln R_i$ 服从负指数分布，因此，单位时间内出现事件的次数 x 由

$$\sum_{i=0}^{x} y_i \leqslant 1 < \sum_{i=0}^{x+1} y_i$$

产生。这里，x 服从泊松分布。具体计算步骤如下：

①根据平均间隔时间，计算 $e^{-\lambda}$。

②置 $i=0$，$S=1$。

③产生伪随机数 R_i。

④计算 $S = S \cdot R_i$。

⑤判别 $S < e^{-\lambda}$。若满足，转⑦，否则转下一步。

⑥$i = i+1$，转③。

⑦$x = i-1$，是泊松分布随机变量的值。其均值为 λ。

（2）二项分布

在产品检验试验中，产品 n 件，合格率 p，废品率 $q = 1-p$，抽取 r 件是合格品的概率是

$$p_r = C_n^r p^r q^{n-r} \qquad r = 0,1,2,\cdots,n \tag{7-22}$$

其分布函数是累积函数。

$$F(x) = \sum_{r=0}^{x} p_r \tag{7-23}$$

可以看到 $F(x)$ 是一个分段连续的函数。即

$$F(r-1) < F(x) \leqslant F(r)$$

成立则 $x = r$。

若 R 为（0，1）上均匀随机数，若有

$$F(r-1) < R \leqslant F(r)$$

则可取 $x = r$。

采用组合法产生二项分布随机变量的值时，计算步骤如下：

①确定二项分布中合格品概率 P。

②产生 m 个（0，1）间均匀随机数 $R_i(i = 1,2,\cdots,m)$。

③对每个 R_i，检验 $R_i \leqslant P$。若满足则 $R_i = 1$，否则，$R_i = 0$。

④计算 $n = \sum_{i=1}^{m} R_i$。n 即为二项分布随机变量的值。

7.2.2.4 近似法

近似法用于分布函数较复杂，不易求解时，下面介绍正态分布的几种近似法。

（1）利用中心极限定理近似

根据中心极限定理，n 个相互独立的随机变量 X_1、X_2、…、X_n，它的均值为 a_i，方差为 σ_i^2，令：

$$X = \sum_{i=1}^{n} X_i ; a = \sum_{i=1}^{n} a_i ; \sigma^2 = \sum_{i=1}^{n} \sigma_i^2 \tag{7-24}$$

若

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} E\{|X_i - E(X_i)|^3\}}{\sigma^3} = 0 \tag{7-25}$$

则随机变量 $Y = \dfrac{X-a}{\sigma}$ 的分布趋向于标准正态分布 $N(0,1)$。

对于 n 个（0，1）上均匀分布的随机数 X_1，…，X_n，其均值 $a_i = \dfrac{1}{2}$，方差 $\sigma_i^2 = \dfrac{1}{12}$。因此，有

$$a = \sum_{i=1}^{n} a_i = \frac{n}{2} ; \sigma^2 = \sum_{i=1}^{n} \sigma_i^2 = \frac{n}{12} ; \sigma = \sqrt{\frac{n}{12}}$$

$$Y = \sqrt{\frac{12}{n}} \left(\sum_{i=1}^{n} X_i - \frac{n}{2} \right)$$

是渐近于 $N(0,1)$ 的。

通过线性变换 $\eta = \mu + \sigma Y$ 可得到均值为 μ，均方差为 σ 的正态分布随机变量 η。

实际仿真应用时，取 $n = 6$ 已有足够精度。此时计算公式是：

$$\eta = \mu + \sqrt{2}\sigma(x_1 + x_2 + x_3 + x_4 + x_5 + x_6 - 3) \tag{7-26}$$

x_1、…、x_6 是计算机产生的 6 个（0，1）的均匀随机数序列。取 $n = 12$ 时公式近似精度更高。

（2）利用博克斯—米勒（Box－Muller）近似

根据博克斯—米勒的近似方法，考虑两个标准正态随机变量 x_1，x_2 的联合密度函数

$$f(x_1, x_2) = \frac{1}{2\pi} e^{-\frac{1}{2}(x_1^2 + x_2^2)} \tag{7-27}$$

作变量变换：

$$x_1 = r\cos\theta ; x_2 = r\sin\theta \tag{7-28}$$

雅可比（Jacobi）行列式 $\left| \dfrac{\partial f(x_1, x_2)}{\partial(r,\theta)} \right| = r$，得：

$$\frac{1}{2\pi}e^{-\frac{1}{2}(r^2\sin^2\theta+r^2\cos^2\theta)}rdrd\theta=\frac{1}{2\pi}e^{-\frac{1}{2}r^2}rdrd\theta=\frac{1}{2\pi}d\theta e^{-\frac{1}{2}r^2}d\left(\frac{1}{2}r^2\right) \tag{7-29}$$

即 θ 是 $[0,2\pi]$ 上的均匀随机变量，$\frac{1}{2}r^2$ 是负指数分布随机变量。故有：

$$F(\theta)=\frac{\theta}{2\pi};F\left(\frac{1}{2}r^2\right)=1-e^{-\frac{1}{2}r^2} \tag{7-30}$$

用逆变换法求得：

$$\theta=2\pi R_2;r=\sqrt{-2\ln R_1}$$

式中，R_2、R_1 是（0，1）均匀随机数。

由此，求得近似公式：

$$x_1=r\cos\theta=\sqrt{-2\ln R_1}\cos(2\pi R_2);x_2=r\sin\theta=\sqrt{-2\ln R_1}\sin(2\pi R_2) \tag{7-31}$$

因此，由产生的随机数 R_1、R_2，按公式（7－31）可得到两个标准正态分布随机变量的值。

（3）多项式近似

多项式逼近的方法有多种，较常用的一种方法是卡恩（Kahn）方法。

对 $N(0,1)$ 的密度函数 $f(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}$ 用下式近似：

$$f(x)=\frac{2Ke^{-Kx}}{1+e^{-Kx}},K=\sqrt{\frac{8}{\pi}}$$

则有分布函数

$$F(x)=\frac{1-e^{-Kx}}{1+e^{-Kx}}$$

采用逆变换法，得

$$x=\frac{1}{K}\ln\left(\frac{1+R}{1-R}\right)$$

R 是（0，1）均匀随机数。

（4）查表法近似

查表法是用一系列直线段（即分段线性函数）来代替标准正态分布函数，而对得到的分段线性近似的分布函数用逆变换法。表 7－2 是产生标准正态分布的变量表。

表 7－2　产生标准正态分布的变量表

输　入 r_i	输　出 x_i	斜　率 a_i	输　入 r_i	输　出 x_i	斜　率 a_i
0	－5.0	33,333	0.57926	0.2	2.63
0.00003	－4.0	756	0.65542	0.4	2.84
0.00135	－3.0	206	0.72575	0.6	3.21
0.00621	－2.5	30.2	0.78814	0.8	3.76
0.02275	－2.0	11.3	0.84134	1.0	4.59
0.06681	－1.5	6.22	0.88493	1.2	6.22
0.11507	－1.2	4.59	0.93319	1.5	11.3
0.15866	－1.0	3.76	0.97725	2.0	30.2
0.21186	－0.8	3.21	0.99379	2.5	206
0.27425	－0.6	2.84	0.99865	3.0	756
0.34458	－0.4	2.63	0.99997	4.0	33,333
0.42074	－0.2	2.52	1.0	5.0	—
0.5	0	2.52			

查表法先把 r_i、x_i、a_i 放在计算机的相应存贮单元中，然后产生一个随机数 R，判别 R 所落入的区间。根据该区间的 r_i、x_i、a_i 按线性插值公式

$$X = x_i + a_i(R - r_i) \tag{7-32}$$

计算标准正态分布随机变量的值。

〔例 7-4〕利用查表法产生正态分布随机变量的值。

若产生的随机数序列是：0.1636，0.9040，0.1871，0.7824，0.5905。以 $R=0.1636$ 为例说明查表法的计算过程。

$R=0.1636$ 落在 $i=8$ 区段，即 $r_8 \leqslant R \leqslant r_9$，查表可得：$x_8=-1.0$；$a_8=3.76$；$r_8=0.15866$。因此，有

$$X=-0.9814256$$

同理，可得其余几个值是：1.3186154，-0.8930656，0.7818465，0.2295612。

查表法的缺点是输入的随机数在接近 0 或 1 时，由于斜率较大，相对误差也较大。虽然落在该区段的随机数的概率不高。

可以看到，若分割的区段数越多，则用于查找的时间就越长。因此，应根据所需分布函数的准确度要求和机时，作折衷的考虑。

查表法还适用于根据抽样数据所得实际系统分布函数的情况。即根据抽样数据建立系统模型，然后用查表法对系统进行仿真研究。

7.2.3 多维随机变量的产生

对于多维随机变量的数值也可以仿照上述的方法，如舍选法、逆变换法等得到。下面对适用于一般情况的条件密度法作简单介绍。

7.2.3.1 条件密度法

若 n 维随机变量 $X=[X_1, X_2 \cdots, X_n]^T$ 的联合分布密度函数是 $f_n(x_1, x_2, \cdots, x_n)$，根据条件概率密度函数的定义，在条件 $X_1=x_1$，$\cdots$，$X_{n-1}=x_{n-1}$ 时 X 的条件概率密度函数是：

$$f(x_n \mid x_1, x_2, \cdots, x_{n-1}) = \frac{f_n(x_1, x_2, \cdots, x_n)}{f_{n-1}(x_1, x_2, \cdots, x_{n-1})}$$

由此，可进行递推得到：

$$f_n(x_1, x_2, \cdots, x_n) = f(x_n \mid x_1, \cdots, x_{n-1}) f(x_{n-1} \mid x_1, \cdots, x_{n-2}) \cdots f(x_2 \mid x_1) f(x_1) \tag{7-33}$$

根据式 (7-33)，可以得到 n 维随机变量数值产生的计算步骤：

①$J=0$。

②$i=1$。

③用一维方法，产生随机数 $x_i^{(J)}$。

④以 $x_1^{(J)}, \cdots, x_{k-1}^{(J)}$ 代入 $f(x_k \mid x_1^{(J)}, x_2^{(J)}, \cdots, x_{k-1}^{(J)})$。

⑤由 $f(x_k \mid x_1^{(J)}, x_2^{(J)}, \cdots, x_{k-1}^{(J)})$ 求得 $x_k^{(J)}$。

⑥判别 $k<n$，若满足，则 $k=k+1$，转④，否则转⑦。

⑦得到 $X^{(J)}=[x_1^{(J)}, \cdots, x_n^{(J)}]^T$。

⑧抽样数 $m>J$ 判别。若满足，则 $J=J+1$，转②。

⑨得 m 组 $X^{(0)}, \cdots, X^{(m-1)}$。

7.2.3.2 正态分布随机向量的产生

设 $\boldsymbol{X}=[X_1, X_2, \cdots, X_n]^T$ 是 n 维正态分布的随机向量。它的均值 $\boldsymbol{\alpha}=E[\boldsymbol{X}]$。协方差

矩阵 $\boldsymbol{\sigma}_x$。

$$\boldsymbol{\sigma}_X = E[(\boldsymbol{X}-\boldsymbol{\alpha})(\boldsymbol{X}-\boldsymbol{\alpha})^T] = \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix} \tag{7-34}$$

其中

$$b_{ij} = E[(X_i - a_i)(X_j - a_j)^T] = b_{ji}$$

因此，$\boldsymbol{\sigma}_x$ 是对称正定阵。

若 $\boldsymbol{\alpha}=0$，σ_x 是 n 阶单位阵，则随机向量 $\boldsymbol{X}$ 是各个分量相互独立的标准正态分布，用 $\boldsymbol{U}$ 表示。通常，存在下列变换关系：

$$\boldsymbol{X} = \boldsymbol{AU} + \boldsymbol{\alpha} \tag{7-35}$$

因此，均方差矩阵

$$\begin{aligned}\boldsymbol{\sigma}_x &= E[(\boldsymbol{X}-\boldsymbol{\alpha})(\boldsymbol{X}-\boldsymbol{\alpha})^T] = E[(\boldsymbol{AU})(\boldsymbol{AU})^T] = E[\boldsymbol{AUU}^T\boldsymbol{A}^T] \\ &= \boldsymbol{A}E[\boldsymbol{UU}^T]\boldsymbol{A}^T = \boldsymbol{AA}^T\end{aligned} \tag{7-36}$$

若 $\boldsymbol{\sigma}_x$ 已知，则 $\boldsymbol{A}$ 可由（7-36）式求得。

以 $n=2$ 为例，说明两维正态分布随机变量 X_1、X_2 的计算方法。

由于 $\boldsymbol{\sigma}_x$ 是正定阵，$\boldsymbol{\sigma}_x = \boldsymbol{AA}^T$，故 $\boldsymbol{A}$ 是下三角阵。即有：

$$\boldsymbol{\sigma}_x = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} a_{11} & a_{21} \\ 0 & a_{22} \end{bmatrix} = \begin{bmatrix} a_{11}^2 & a_{11}a_{21} \\ a_{11}a_{21} & a_{21}^2 + a_{22}^2 \end{bmatrix} \tag{7-37}$$

得 a_{ij} 为：

$$a_{11} = \sqrt{b_{11}}; a_{21} = \frac{b_{21}}{\sqrt{b_{11}}}; a_{22} = \sqrt{\frac{b_{11}b_{22} - b_{12}^2}{b_{11}}} \tag{7-38}$$

由此，得两维随机变量抽样公式：

$$X_1 = \sqrt{b_{11}}u_1 + a_1 \tag{7-39}$$

$$X_2 = \frac{1}{\sqrt{b_{11}}}\left(b_{21}u_1 + \sqrt{b_{11}b_{22} - b_{12}^2}u_2\right) + a_2 \tag{7-40}$$

类似地，对于 n 维正态分布随机变量的抽样公式，计算步骤如下：

①产生 n 个独立的 $N(0,1)$ 随机数 u_1，u_2，…，u_n。

②按下列递推公式计算 η'_k：

$$\eta'_1 = \sigma u_1 \tag{7-41}$$

$$\eta'_k = e^{-\alpha}\eta'_{k-1} + \sigma\sqrt{1-e^{-2\alpha}}u_k \qquad 2 \leqslant k \leqslant n \tag{7-42}$$

③计算 X_k：

$$X_k = \eta'_k + a_k \qquad 1 \leqslant k \leqslant n \tag{7-43}$$

7.2.4 随机过程的产生

7.2.4.1 一维正态平稳过程

设一维正态分布的平稳过程 $X(t)$ 具有均值 $a_x(t)=0$，协方差 $\sigma_x(t)$。产生该过程的方法是在有限个离散时间点上产生可能取值 $X(k\Delta t)$。

(1) $\sigma_x(t) = \sigma^2 e^{-\alpha|t|}$

根据正态随机向量的产生方法，有：

$$b_{ij} = \sigma^2 e^{-\alpha(i-j)\Delta t} \tag{7-44}$$

因此，可按上述方法直接得到 $X(t)$。其方法如下：

①产生 n 个独立的 $N(0,1)$ 随机数 u_1，…，u_n。

② 计算：$X_1 = \sigma u_1$ (7-45)

$$X_k = X(k\Delta t) = e^{-\alpha\Delta t}X_{k-1} + \sigma\sqrt{1-e^{-2\alpha}}u_k \quad 2 \leqslant k \leqslant n \tag{7-46}$$

式中 $\Delta t = t_k - t_{k-1}$

(2) $\sigma_x(t) = \sigma^2\cos\beta t e^{-\alpha|t|}$

若 $X_1(t), X_2(t)$ 是两个独立的正态平稳过程。其均值为 0，协方差 $\sigma^2 e^{-\alpha|t|}$。则

$$X(t) = X_1(t)\cos\beta t + X_2\sin\beta t \tag{7-47}$$

组成的过程是均值为 0，协方差为 $\sigma^2\cos\beta t e^{-\alpha|t|}$ 的正态平稳过程。

因此，可以先用（1）的方法产生两个独立的正态平稳过程，然后，按（7－47）计算 $X(t)$。

7.2.4.2 n 维正态马尔柯夫过程

可以按条件均值和条件协方差阵的方法产生。以二维正态马尔柯夫过程为例说明如下：

要求产生均值 $a(t) = 0$，协方差阵 $\sigma_x(t,\tau)$ 为：

$$\sigma_x(t,\tau) = \begin{pmatrix} 2e^{-\beta|t-\tau|} & e^{-\beta|t-\tau|} \\ e^{-\beta|t-\tau|} & e^{-\beta|t-\tau|} \end{pmatrix} \quad \beta > 0 \tag{7-48}$$

二阶正态马尔柯夫过程的条件期望是：

$$\begin{aligned} a[t \mid X(\tau)] &= E[X(t) \mid X(\tau) = X(\tau)] \\ &= a(t) + \sigma_x(t,\tau)\sigma_x^{-1}(\tau,\tau)\sigma_x(\tau,t) \\ &= e^{-\beta|t-\tau|}\begin{pmatrix} x_1(\tau) \\ x_2(\tau) \end{pmatrix} \end{aligned} \tag{7-49}$$

条件协方差矩阵 $\sigma_x[t \mid X(\tau)]$ 是：

$$\sigma_x[t \mid X(\tau)] = (1 - e^{-2\beta|t-\tau|})\begin{Bmatrix} 2 & 1 \\ 1 & 1 \end{Bmatrix} \tag{7-50}$$

由此，求得 A 的各元素：

$$A = \begin{pmatrix} a_{11} & 0 \\ a_{21} & a_{22} \end{pmatrix} = \left(\frac{1-e^{-2\beta|t-\tau|}}{2}\right)^{\frac{1}{2}}\begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix}$$

本例的计算步骤是：

①产生两上互相独立的 N（0，1）随机向量 $\begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix}$，$\begin{pmatrix} u_{21} \\ u_{22} \end{pmatrix}$，…，在 $t_1 < t_2 < \cdots$ 上产生随机过程 $X(t)$。

②计算随机过程的时间序列：

$$\begin{pmatrix} X_1(t_1) \\ X_2(t_1) \end{pmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}\begin{pmatrix} u_{11} \\ u_{12} \end{pmatrix} \tag{7-51}$$

$$\begin{pmatrix} X_1(t_k) \\ X_2(t_k) \end{pmatrix} = e^{-\beta\Delta t}\begin{pmatrix} X_1(t_{k-1}) \\ X_2(t_{k-1}) \end{pmatrix} + A\begin{pmatrix} u_{k1} \\ u_{k2} \end{pmatrix} \quad 2 \leqslant k \leqslant n \tag{7-52}$$

式中，t_1 是初始时刻，$\Delta t = t_k - t_{k-1}$，通常时间间隔取相等，但也可取不等距时间间隔。

7.2.5 数字仿真子程序

7.2.5.1 随机数发生器数字仿真子程序

在 BASIC 语言中，提供了（0，1）均匀分布随机函数 RND（ ），可供调用。例如用

$$RN = RND(1) * (B - A) + A$$

可以产生（A,B）内均匀分布随机数。图 7－3 是由标准随机函数产生的随机数的分布。

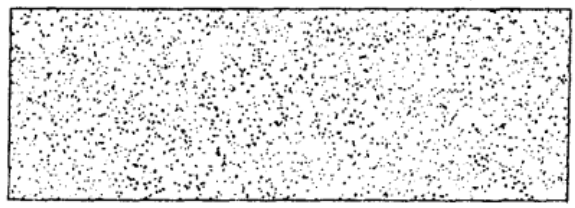


图 7－3 随机数的分布

GPSS 语言中提供了混合同余法产生伪随机数的 FORTRAN 程序。对于 32 位字长的计算机采用模 $M = 2^{30}$ 及 $C = 227623267$，而 A 可取 30 个值之一。用 BASIC 语言编写时，采用双精度格式。其子程序包括初始化子程序及产生伪随机数子程序两部分。子程序清单如下：

```
1000  REM Initialization
1002  FOR I=1 TO 30
1004  READ RA#(I)
1006  RC#(I)=227623267#
1008  DR#(I)=1
1010  NEXT I
1012  MODM#=1073741824#: REM 2^30
1014  RETURN
1016  DATA 10753813, 228181237, 348984893, 590634277, 10841757
1018  DATA 107408213, 469770781, 832159853, 10842045, 228244373
1020  DATA 348969437, 10783125, 107452989, 469773661, 590648085
1022  DATA 10784189, 228206429, 469793853, 832230813, 107464661
1024  DATA 469808301, 590585901, 228178005, 349045949, 10845149
1026  DATA 107441485, 469795933, 349062285, 107380645, 10767581
2000  REM RANDOM NUMBER
2002  DR#(NUM)=DR#(NUM)*RA#(NUM)+RC#(NUM)
2004  IF DR#(NUM)<MODM THEN 2008
2006  DR#(NUM)=DR#(NUM)-MODM#*INT(DR#(NUM)/MODM
#)
2008  RN#=DR#(NUM)/(MODM#-1)
2010  RETURN
```

图7－4 是上述子程序产生的伪随机数的分布。采用的是混合同余法。其中，a = 10753813。

在实际应用时，可以上述两个方法组合应用。即用 RND（·）产生（0，30）间随机数，然后，把此数作为 NUM，代入混合同余法算式，得到均匀随机数。

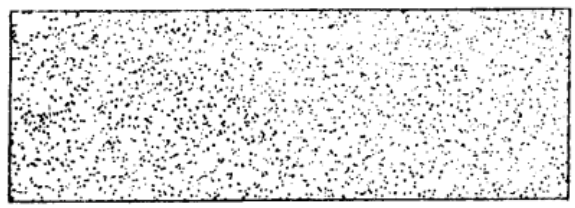


图 7－4　混合同余法产生的随机数分布

7.2.5.2　随机分布的数字仿真子程序

1．正态分布

下面是利用博克斯—米勒近似方法得到正态分布的仿真子程序。

```
2100  REM NORMAL DISTRIBUTION
2102  NUM = RND (1) * 30
2104  GOSUB 2000
2105  IF RN# = 0 THEN 2102
2106  R# = RN#
2108  NUM = RND (1) * 30
2110  GOSUB 2000
2112  A# = SQR (-2 * LOG (R#))
2114  B# = 2 * 3.1415926535# * RN#
2116  V1# = A# * COS (B#)
2118  V2# = A# * SIN (B#)
2120  U1# = V1# * SIGMA + MEAN
2122  U2# = V2# * SIGMA + MEAN
2124  RETURN
```

在程序中，采用组合的方法产生随机数。子程序得到的 V1#，V2# 是标准正态分布（0，1），而 U1#，U2# 是均值为 MEAN，均方差为 SIGMA 的正态分布。

图 7－5 就是根据上述子程序得到的正态分布。图 7－6 是相应的实际仿真结果绘制的正态分布函数。

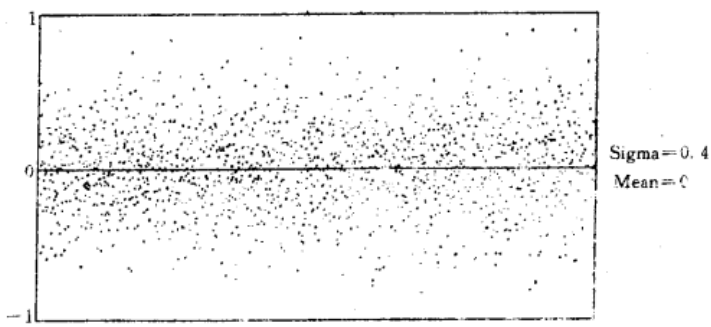


图 7－5　正态分布的抽样值

2．爱尔朗分布（Erlang）

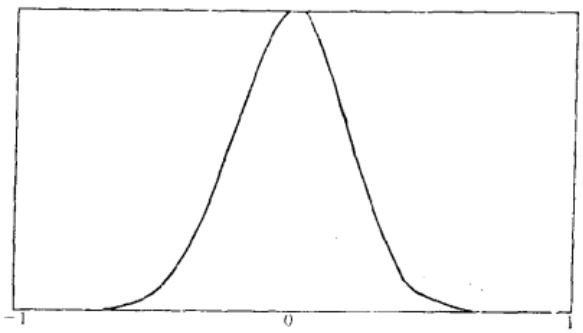


图 7-6　实测正态分布函数

爱尔朗分布是离散事件系统中的常见分布之一。若每个顾客的服务时间相互独立，服从同一负指数分布，则 k 个顾客所需服务时间服从爱尔朗分布。若 k 个服务台串联，每台服务时间相互独立，均为同一负指数分布，则每个顾客总的服务时间也服从爱尔朗分布。

爱尔朗分布的分布密度函数是：

$$f(x)=\frac{\lambda^k x^{k-1}}{(k-1)!}e^{-\lambda x} \tag{7-53}$$

根据爱尔朗到达分布和 k 级指数到达分布是等效的特性，若指数分布均值为$\frac{1}{\lambda}$，则爱尔朗分布的随机变量均值为$\frac{k}{\lambda}$，因此，得：

$$x=-\frac{1}{\lambda}\ln\prod_{i=1}^{k}u(i) \tag{7-54}$$

$u(i)$ 是(0,1) 均匀随机数。可以看到，$k=1$ 即为负指数分布。

下面是爱尔朗分布的仿真子程序：

```
2200  REM ERLANG DISTRIBUTION
2205  IF K>=1 THEN 2215
2210  RETURN
2215  ALF=K/MEAN
2220  R#=1
2225  FOR I=1 TO K
2230  NUM=RND (1) *30
2235  GOSUB 2000
2240  R#=R#*RN#
2245  NEXT I
2250  IF R#=0 THEN 2220
2255  U1#=-1/ALF*LOG (R#)
2260  RETURN
```

7.3 离散事件系统的数字仿真

7.3.1 离散事件系统的数学模型

离散事件系统是包含随机变量的随机系统。与连续系统数字仿真类似，离散事件系统的数字仿真也需为真实系统建立仿真模型。对于确定性系统，根据系统状态和输入之间的函数关系、状态初值，可以递推计算相应时刻的状态，并计算系统输出。对于随机系统，由于计算机产生的随机变量采样值的算法是确定性算法，因此，其模型也是确定性的。而与确定性系统数字仿真的区别则是在对随机系统的数字仿真中，赋以了概率测度。

7.3.1.1 仿真模型的建立

以仪修车间仿真实例说明离散事件系统仿真模型建立的方法。

若某厂仪表修理车间的仪表调试装置每次只能调试一台仪表。送修仪表到达时，若调试装置空闲，就直接送入接受修理，修理后则出口处取走。若送修仪表到达时，调试装置在使用，则只能放入等待区，按先入先修的规律等待修理。若等待区足够大，则待修仪表占满可不考虑。

经使用，一年统计表明，仪表平均等待时间达 60 天，平均逗留时间为 75 天，调试仪表装置利用率为0.98。为了了解若增加一台调试装置后，调试装置的利用率，需进行仿真研究。

(1) 系统组成成分

系统组成成分是指对仿真有影响的一些实体。它的行动会影响系统的输出。系统的组成成分常用随机变量来描述。

本例中，入口、等待区和修理区是系统的三个组成成分。

(2) 描述变量

描述变量是系统各实体的属性。包括系统的内部变量和外部变量。与确定性系统相似，可分为输入、输出和状态变量。

本例中，用 $(u_i, T_{1,i})$ 二维变量表示仪表 u_i 于 $T_{1,i}$时刻到达入口。排队等待的队列是 $Q_1Q_2\cdots Q_M$，队长是 M。用 $(x_j, T_{1,j})$ 表示仪表 x_j 于 $T_{1,j}$时刻送上调试台接受修理。用 $(x_j, T_{2,j})$ 表示仪表 x_j 于 $T_{2,j}$时刻修好，并离去。

因此，可列出输入变量是：

$$(u_i, T_{1,i}),\ T_{1,i} \in [0, 365] \tag{7-55}$$

状态变量

$$S = \begin{Bmatrix} Q_1Q_2\cdots Q_M \\ (x_j, T_{2,j}) \end{Bmatrix},\ T_{2,j} \in [0, 365] \tag{7-56}$$

输出变量为总等待时间 TWT，总空闲时间 TFT，及送修仪表总件数 NT。

(3) 相互关系

相互关系指出各组成成分变量间的相互关联。它是系统描述的重要内容。

本例中，两台相邻送修仪表的到达间隔时间

$$DT_1 = T_{1,i+1} - T_{1,i} \tag{7-57}$$

是一个随机变量。经一年观察统计，其密度函数服从截尾指数分布。即：

$$f^*(DT_1)=\begin{cases} f(DT_1)\Big/\int_{DT_{1\min}}^{DT_{1\max}} f(DT)\,dDT & DT_{1\min}<DT_1\leqslant DT_{1\max} \\ 0 & DT_1>DT_{1\max}\text{ 或 }DT_1\leqslant DT_{1\min} \end{cases} \tag{7-58}$$

式中，$f(DT_1)=\frac{1}{TA}e^{-\frac{1}{TA}DT_1}$ 为末截尾负指数分布。TA 是 DT_1 均值。$DT_{1\min}$、$DT_{1\max}$是截尾区间下限和上限。

调试修理一台仪表所需时间 DT_2 也服从截尾指数分布。其均值为 TS，截尾区间上、下限为 $DT_{2\max}$、$DT_{2\min}$。

设当前时刻是 T，它的离散值与当前状态变化时刻相一致。则通过随机采样 DT_1、DT_2，可得：

$$T_{1,i+1}=T+DT_1;T_{2,j+1}=T+DT_2 \tag{7-59}$$

当前时刻状态 S_k 与下一生成时刻状态S_{k+1}之间的转移关系是：

$$\begin{aligned} S_{k+1}&=\delta[S_i,(u_i,T_{1,i})_{k+1}] \\ &=\delta\left\{\begin{bmatrix}Q_1\cdots Q_M\\(x_j,T_{2,j})\end{bmatrix}_k,(u_1,T_{1,i})_{k+1}\right\} \end{aligned} \tag{7-60}$$

而 TWT、TFT、NT 等可由下列关系计算：

$$\begin{bmatrix}NT_{i+1}\\TWT_{k+1}\\TFT_{j+1}\end{bmatrix}=\begin{bmatrix}NT_i+1\\TWT_k+M*(T_{k+1}-T_k)\\TFT_j+(T_{j+1}-T_j)\end{bmatrix} \tag{7-61}$$

（4）列出仿真模型程序流程图。编制程序

本例的计算步骤如下：

①置初值。

②生成下一时刻。

③检查是否到达仿真终止时间。满足转⑧。

④进行状态转移的有关计算。

⑤计算输出变量。

⑥仿真时钟前进一步。转②。

⑦计算其他输出变量。

⑧结束。

据此，可编制仿真程序并运行。

7.3.1.2 仿真模型的三要素

通过上面的例子，可以看到离散事件系统的仿真模型通常包括下列三要素。

（1）到达模型

到达模型描述系统中临时实体到达系统的时间特性。

两相邻实体到达系统的间隔时间 DT_1 是一个随机变量，用到达分布函数 $A_0(t)$ 表示。由于在任一时间区间内到达的概率正比于间隔时间，而与当前时刻无关。因此，$A_0(t)$ 可用负指数分布描述。即：

$$A_0(t)=e^{-\lambda t} \tag{7-62}$$

式中，$\lambda = \frac{1}{DT_1}$ 是到达速率。上面的例子中的统计结果表明这种描述是符合实际情况的。

在给定时间 t 内，发生 n 个临时实体到达的模型可用泊松分布描述，分布密度函数是：

$$f(x) = \frac{\lambda^x}{x!}e^{-\lambda}$$

仿真时，若已知平均间隔时间，则按泊松分布的到达模型可以方便地计算出相邻事件的间隔时间。

（2）服务模型

服务模型描述了永久实体为临时实体服务的时间模型。如上例中的调试仪表时间 DT_2。

一般来说，若有 n 个永久实体，永久实体为单个临时实体服务所需时间为 T_s，则用服务分布函数 $S_0(t)$ 来描述随机变量 T_s。

在上例中，$S_0(t)$ 服从负指数分布。一般情况，$S_0(t)$ 是均值为 T_s 的均匀分布。

（3）排队规则

排队规则是服务过程中，出现排队时的处理规则。当永久实体服务速度低于临时实体互相到达速度时，在永久实体前就出现临时实体的排队。不出现排队说明服务系统效率低，这就需要调整永久实体数量等。同样，排队时间过长也需要对服务系统进行调整。排队规则是在临时实体组成的等待服务队列中挑选下一个接受服务的临时实体的规则。排队规则通常有：

①先进先出（FIFO）

这是队列服务方式，采用先进入等待区的临时实体先得到服务的排队规则。

②后进先出（LIFO）

这是堆栈服务方式，采用后进入堆栈的临时实体先弹出接受服务的排队规则。

③随机服务

这是竞争服务方式，采用等待的临时实体不断询问永久实体是否可提供服务，一旦某永久实体空闲则可对其实施服务的排队规则。

④优先级

对临时实体实施优先级别高的先得到服务的排队规则。

7.3.1.3 流程控制

对离散事件系统仿真时，仿真模型还应包括对流程的控制功能，主要是时间进程管理、通道选择和同时事件的管理等。

（1）仿真时钟

仿真时钟表示仿真运行开始到仿真某时刻共花费多少时间。要区分仿真时间和计算机仿真所花费机时、仿真时间和实际时钟时间。

（2）时间进程管理和通道选择

时间进程管理是用于控制仿真过程运行进程的。常用两种方法。

定步长法　把整个仿真过程分为许多等间隔的时间区段，间隔的长度可根据具体系统确定。系统的仿真进程是先把系统初值送入计算机，然后，每前进一步，就对系统所有实体、描述变量等进行一次全面的扫描，若发生某一事件，就转相应子程序进行事件处理，直到全部扫描完，再把仿真钟步进一步，直到仿真时间结束。

事件表法　事件表法以事件为线索，每一次步长是相邻两事件的间隔时间，它是一种活

动所持续的过程。每前进一步，系统的状态就发生一次转移。可以看到，事件表法与定步长法有下列几点不同：

①事件表法不是等步长步进的，它的时间序列是不规则的离散点，而系统的状态变量仅在这些离散点取值并发生变化。

②定步长法由于不能区分在同一步内不同时发生的若干事件，从而引入仿真误差。而事件表法的每一事件都有确切的时间相对应，不会产生误差。此外，对同一系统进行相同时间的仿真结果表明，事件表法精确度较高。

③定步长法每步进一步就要对整个系统进行全面的扫描，而事件表法每步进一步只对一个实体活动进行处理。因此，当两相邻事件发生的平均间隔时间较长时，事件表法可以节省机时。

主要通道选择方法与时间管理方法联系在一起。事件表法中，由于时间的推进点发生在某事件的发生时刻，因此，通道就选择指向该事件。而定步长法中，一旦推进进程中发现已有事件生成，则根据有关排队规则对通道进行选择，并转向相应通道进行处理。

(3) 同时事件管理

同一时刻发生的几个事件称为同时事件。根据同时发生事件是否属于同一类型，可分为同类同时事件、混合同时事件。对同时事件的处理，需解决处理顺序问题。一般可以按扫描的先后顺序进行处理。若要求按照某种先后顺序处理时，应列出优先顺序表，然后依顺序处理。

7.3.2 离散事件系统的仿真策略和结构模型

仿真策略就是系统仿真进程的调度方法。它是企图使模型描述的某些形式比其他形式更易表示和能被更有效地进行处理的方法。三种通用的仿真策略是：事件预定、活动扫描和进程互配。

7.3.2.1 面向事件结构模型

面向事件结构模型是按事件预定的仿真策略组建的。它强调预定全部事件，不采用对全部实体扫描的方法。因此状态的变化发生在预定的事件生成时刻。采用事件表法进行仿真。下面以计算机与用户终端系统为例说明。

该系统有计算机系统和 90 个用户和工作人员组成。用户写好程序到用户终端调试，调试结束后，程序送入中央处理单元（CPU），按照先入先出排队规则（组成用户队）等待处理。CPU 处理完前一程序后就对用户队首的程序处理。处理完后的计算结果放入打印结果队（批次队），按每 8 小时，由工作人员打印一次，交用户。用户拿到计算结果离开系统。

这里，用户写、调程序的时间是随机变量。用户程序由计算机运行也是随机变量。下面列出有关步骤。

(1) 组成成分

主动成分：用于主动产生状态变化的组成成分。本例是用户 1、用户 2、…、用户 90、计算机、工作人员。

被动成分：在主动成分作用下产生状态变化的组成成分。本例是用户队、批次队。

(2) 描述变量

①描述用户 i 的变量（$i = 1, \cdots, 90$）

用户状态 S_i，值域 $\{0, 1, 2\}$。

用户下一生成时刻 t_i，值域 $\{R_{0,\infty}^{+}\}$。

用户写、调程序时间 WCT，取决于用户 i 类型（TYPE(i)）的随机变量。

②描述计算机的变量

CPU 状态 S_{CPU}，值域 $\{0, 1\}$。

CPU 下一生成时刻 t_{CPU}，值域 $\{R_{0,\infty}^{+}\}$。

CPU 工作时间 WT，取决于用户 i 类型的随机变量。

③描述工作人员的变量。

下一生成时刻（即下一次批处理时刻）t_{CL}。

④描述用户队的变量

用户队状态 S_{UQ}，值域 $\{0,1,2,\cdots,90\}^{*}$，（A^{*}代表由集合中元素组成的全部有限序列集合）。

⑤描述批次队的变量

批次队的状态 S_{BQ}，值域 $\{0,1,\cdots,90\}$ 的幂集合。

（3）参数

用户类型函数 TYPE(i)。

（4）相互关系

用户 i 状态：

0——写、调程序，时间为 WCT。

1——向 CPU 送入程序 i，即进入用户队，直到 i 处理完毕。然后进入批次队。

2——等在批次队，直到打印输出后转入 0 状态。

计算机状态：

0——CPU 空闲，接受用户程序后，处理程序，花时为 WT，处理毕程序进入批次队。

1——CPU 忙。程序进入用户队的队尾。如 CPU 空闲，用户队空，则计算机每隔 1/10 秒扫描一次，检查用户程序是否到达。

工作人员状态：

每隔 8 小时处理批次队的计算结果。

各主动成分的影响对象：

用户 i：用户队、批次队和用户 i 自己。

计算机：用户队、用户、计算机本身。

工作人员：批次队、用户、工作人员自己。

用户 $i(i = 1,\cdots,90)$ 的局部转移函数

$$\delta_i(S_i, t_i, r_{i,1}, r_{i,2}, S_{UQ}, S_{BQ}) = (S'_i, t'_i, r'_{i,1}, r'_{i,2}, S'_{UQ}, S'_{BQ}) \tag{7-63}$$

式中 $r_{i,1}$——WCT 的种子。$r_{i,2}$——WT 的种子。

状态转移：

$$S'_i = S_{i+1}(\mathrm{mod}\ 3) \tag{7-64}$$

$$t'_i = \begin{cases} t_i + WCT(\mathrm{TYPE}(i), r_{i,1}) & \text{当 } S_i = 2 \text{ 时} \\ \infty & \text{其余情况} \end{cases} \tag{7-65}$$

$$r'_{i,1} = \begin{cases} \Gamma(r_{i,1}) & \text{当 } S_i = 2 \text{ 时} \\ r_{i,1} & \text{其余情况} \end{cases} \tag{7-66}$$

$$r'_{i,2} = r_{i,2} \tag{7-67}$$

$$S'_{UQ}=\begin{cases}S^{i}_{UQ} & \text{当 } S'_i=1\\ S_{UQ} & \text{其余情况}\end{cases} \tag{7-68}$$

$$S'_{BQ}=\begin{cases}S_{BQ}\cup|i| & \text{当 } S'_i=2\\ S_{BQ} & \text{其余情况}\end{cases} \tag{7-69}$$

式中，$WCT(\mathrm{TYPE}(i),r_{i,1})$ 是种子随机数发生器的输出和用户类型的函数。$\Gamma(r_{i,1})$ 是伪随机数发生器的状态转移函数，即 $(r_{i,1})_j$ 是第 j 个随机数时，有 $(r_{i,1})_{j+1}=\Gamma((r_{i,1})_j)$

计算机的局部转移函数：

$$\begin{aligned}&\delta_c(S_1,t_1,r_{1,1},r_{1,2},\cdots,S_{90},t_{90},r_{90,1},r_{90,2},S_{\mathrm{CPU}},t_{\mathrm{CPU}},S_{UQ})\\&=(S'_1,t'_1,r'_{1,1},r'_{1,2},\cdots,S'_{90},t'_{90},r'_{90,1},r'_{90,2},\cdots S'_{\mathrm{CPU}},t'_{\mathrm{CPU}},S'_{\mathrm{UQ}})\end{aligned} \tag{7-70}$$

状态转移：

$$S'_i=S_i,\ i=1,\cdots,90 \tag{7-71}$$

$$t'_i=\begin{cases}t_{\mathrm{CPU}} & \text{当 } i=\mathrm{top}(S_{UQ}) \text{ 且 } S_{\mathrm{CPU}}=1\\ t_i & \text{其余}\end{cases} \tag{7-72}$$

$$S'_{\mathrm{CPU}}=\begin{cases}1 & \text{当 } S_{\mathrm{CPU}}=0 \text{ 且 } S_{UQ}\neq A\\ 0 & \text{其余}\end{cases} \tag{7-73}$$

$$t'_{\mathrm{CPU}}=\begin{cases}t_{\mathrm{CPU}}+0.1 & \text{当 } S_{\mathrm{CPU}}=0 \text{ 且 } S_{UQ}=A\\ t_{\mathrm{CPU}}+\mathrm{WT}(\mathrm{TYPE}(\mathrm{top}(S_{UQ})),r_{\mathrm{top}}(S_{UQ}),2) & \\ \qquad \text{当 } S_{\mathrm{CPU}}=0 \text{ 且 } S_{UQ}\neq A & \\ t_{\mathrm{CPU}} & \text{其余}\end{cases} \tag{7-74}$$

$$S'_{UQ}=\begin{cases}\mathrm{rest}(S_{UQ}) & \text{当 } S_{\mathrm{CPU}}=1\\ S_{UQ} & \text{其余}\end{cases} \tag{7-75}$$

$$r'_{i,1}=r_{i,1} \qquad i=1,2,\cdots,90 \tag{7-76}$$

$$r'_{i,2}=\begin{cases}\Gamma(r_{i,2}) & \text{当 } i=\mathrm{top}(S_{UQ}) \text{ 且 } S_{\mathrm{CPU}}=0\\ r_{i,2} & \text{其余}\end{cases} \tag{7-77}$$

上式中，定义 top 及 rest 函数为：

$\mathrm{top}(a_1a_2\cdots a_n)=a_1$　　取队首元素

$\mathrm{rest}(a_1a_2\cdots a_n)=a_2\cdots a_n$　　取 top 的其余部分

工作人员的局部转移函数：

$$\begin{aligned}&\delta_{CL}(S_1,t_1,r_{1,1},\cdots,S_{90},t_{90},r_{90,1},r_{90,2},S_{BQ},t_{CL})\\&=(S'_1,t'_1,\cdots,S'_{90},t'_{90},r'_{90,1},\cdots,S'_{BQ},t'_{CL})\end{aligned} \tag{7-78}$$

状态转移：

$$S'_i=S_i \qquad (i=1,\cdots,90) \tag{7-79}$$

$$t'_i=\begin{cases}t_{CL} & i\in|S_{BQ}|\\ t_i & \text{其余}\end{cases} \tag{7-80}$$

$$r'_{i,1}=r_{i,1} \tag{7-81}$$

$$r'_{i,2}=r_{i,2} \tag{7-82}$$

$$t'_{CL}=t_{CL}+8\times 60 \tag{7-83}$$

$$S'_{BQ} = A \tag{7-84}$$

7.3.2.2 活动扫描结构模型

活动扫描结构模型是按活动扫描策略组建的。它用扫描总体状态的方法查看有否符合条件的事件可以生成。如有则激发生成，如没有则按下一预定事件生成时刻步进仿真时钟，再在相应时间点进行扫描。因此。它用事件预定策略进行时间管理。但不激发事件的生成。激发事件生成的条件不仅包含时间条件，还包含状态条件。即应满足：

① $t_a \leqslant t$ (7-85)

② $C_a(s) = \text{true}$ (7-86)

式中，①是时间条件，t_a 是成分 a 局部状态的下一生成时刻，t 是当前时刻。$C_a(s)$ 是条件函数。$C_a(s) = \text{true}$ 表示成分 a 在总体状态 s 时状态条件满足，若 $C_a(s) = \text{false}$ 则表示不满足状态条件。

当 $t_a = t$ 满足而生成的事件激发称预定激发，而满足上述两条件的事件激发称条件激发。因此，事件预定策略中是预定激发。

活动扫描策略的步骤是：

①把主动成分 $a \in D_A$ 按它们的下一生成时刻 t_a 与当前时刻 t 分成三类：

$FUTURE(s) = \{a \mid t_a > t\}$

$PRESENT(s) = \{a \mid t_a = t\}$

$PAST(s) = \{a \mid t_a < t\}$

②把满足下列条件

$a \in PRESENT(s) \cup PAST(s)$

$C_a(s) = \text{true}$

的成分放入集合

$$ACT(s) = \left\{ a \,\middle|\, \begin{matrix} a \in PRESENT(s) \cup PAST(s) \\ C_a(s) = \text{true} \end{matrix} \right\} \tag{7-87}$$

中，作为当前时刻 t 的激发事件。

③按优先级顺序选出一个成分

$\bar{a}(s) = SELECT(ACT(s))$ 并执行相应的局部转移函数规定的状态转移。即对 $\bar{a}(s)$ 事件激发。

④计算下一生成时刻 $t'(s)$

当 $ACT(s) \neq A$ 时，$t'(s) = t$

当 $ACT(s) = A$ 且 $FUTURE(s) \neq A$ 时，$t'(s)$ 由

$t'(s) = \min(t_a \mid a \in FUTURE(s))$

决定，仿真时钟 $t = t'(s)$。

⑤重复上述步骤，直到仿真终止时间。

下面把计算机和用户终端系统用活动扫描结构模型来描述。

模型中，计算机是被动成分，仅显示忙或空闲。用户不断查询是否已成用户队的队首，若是，则计算机处理。处理时间为 ST。

用户离开 CPU 由工作人员控制，用户进入批次队，不断查询批次队的门是否打开，若打开则批次队内用户可以出门，并转入初始状态，批次队变成空队。

这里用户对用户队及批次队门的查询组成了活动扫描。工作人员对批次队门的控制也是

不断查询批次队是否空，这也是活动扫描。

下面列出有关步骤：

(1) 组成成分

除了计算机成为被动成分外，与预定事件结构模型相同。被动成分增加批次队门。

(2) 描述变量

①描述用户 i 的变量 ($i = 1, \cdots, 90$)

用户 i 的状态 S_i，值域 {0，1，2，3，4}。

用户 i 的下一生成时刻 t_i，值域 $\{R_{0,\infty}^{+}\}$。

②描述工作人员的变量

工作人员状态 S_{CL}，值域 {0，1}。

工作人员下一生成时刻 t_{CL}，值域 $\{R_{0,\infty}^{+}\}$。

③描述计算机的变量

计算机状态 S_{CPU}，值域 {0，1}。

④描述用户队的变量

用户队状态 S_{UQ}，值域 $\{0,1,2,\cdots,90\}^{*}$。

⑤描述批次队的变量

批次队状态 S_{BQ}，值域 $\{0,1,2,\cdots,90\}$ 的幂集合。

⑥描述批次队门的变量

批次队门状态 S_G，值域 $\{0,1\}$

(3) 参数

TYPE(i)：用户类型函数

(4) 相互关系

用户 i 状态：

0——写、调试程序。花时 WCT。

1——进入用户队。

2——i 成为用户队首且计算机空闲时，接受计算机处理，处理时间 ST。

3——处理后程序 i 进入批次队，等门开。

4——批次队门开，用户 i 出门并转入初始状态。

工作人员状态：

每隔 8 小时开批次队门。

计算机状态：

0——空闲；1——忙。

批次队门状态：0——门开；1——门关。

各主动组成成分的影响对象：

用户 i：计算机、用户队、批次队。

工作人员：批次队门。

用户 i 的条件函数：

$$C_i(s) = C_i(S_i, S_{CPU}, S_{UQ}, S_G)$$

$$= \begin{cases} \text{true} & \text{当 } S_i = 1, i = \text{top}(S_{UQ}) \text{ 且 } S_{CPU} = 0 \text{ 时} \\ & \text{或 } S_G = 0 \text{ 且 } S_i = 3 \text{ 时或 } S_G = 0 \text{ 且 } S_i = 4 \text{ 时} \\ & \text{或 } S_i = 0 \text{ 时或 } S_i = 2 \text{ 时} \\ \text{false} & \text{其余} \end{cases} \tag{7-88}$$

局部转移函数：

$$\delta_i((S_i, t_i, r_{i,1}, r_{i,2}), S_{CPU}, S_{UQ}, S_{BQ}, S_G) = ((S'_i, t'_i, r'_{i,1}, r'_{i,2}), S'_{CPU}, S'_{UQ}, S'_{BQ}) \tag{7-89}$$

状态转移：

$$S'_i = S_{i+1} \pmod 5 \tag{7-90}$$

$$t'_i = \begin{cases} t + WCT(\text{TYPE}(i), r_{i,1}) & \text{当 } S_i = 4 \text{ 时} \\ t + ST(\text{TYPE}(i), r_{i,2}) & \text{当 } S_i = 1 \text{ 时} \\ t_i & \text{其余} \end{cases} \tag{7-91}$$

$$r'_{i,1} = \begin{cases} \Gamma(r_{i,1}) & \text{当 } S_i = 4 \text{ 时} \\ r_{i,1} & \text{其余} \end{cases} \tag{7-92}$$

$$r'_{i,2} = \begin{cases} \Gamma(r_{i,2}) & \text{当 } S_{i=1} \text{ 时} \\ r_{i,2} & \text{其余} \end{cases} \tag{7-93}$$

$$S'_{CPU} = \begin{cases} 0 & \text{当 } S_{UQ} = A \text{ 或 } S_i = 2 \text{ 时} \\ 1 & \text{其余} \end{cases} \tag{7-94}$$

$$S'_{UQ} = \begin{cases} S^i_{UQ} & \text{当 } S_i = 0 \text{ 时} \\ \text{rest}(S_{UQ}) & \text{当 } S_i = 2 \text{ 时} \\ S_{UQ} & \text{其余} \end{cases} \tag{7-95}$$

$$S'_{BQ} = \begin{cases} S_{BQ} \cup \{i\} & \text{当 } S_i = 2 \text{ 时} \\ S_{BQ} - \{i\} & \text{当 } S_i = 3 \text{ 时} \\ S_{BQ} & \text{其余} \end{cases} \tag{7-96}$$

工作人员条件函数

$$C_{CL}(S_{CL}, S_{BQ}) = C_{CL}(s) = \begin{cases} \text{true} & \text{当 } S_{CL} = 1 \text{ 且 } S_{BQ} = A \text{ 时或 } S_{CL} = 0 \text{ 时} \\ \text{false} & \text{其余} \end{cases} \tag{7-97}$$

工作人员局部转移函数

$$\delta_{CL}((S_{CL}, t_{CL}), S_{BQ}, S_G) = ((S'_{CL}, t'_{CL}), S'_G \tag{7-98}$$

状态转移：

$$S'_{CL} = S_{CL} + 1 \pmod 2 \tag{7-99}$$

$$t'_{CL} = \begin{cases} t_{CL} & \text{当 } S_{CL} = 0 \text{ 时} \\ t_{CL} + 8 \times 60 & \text{当 } S_{CL} = 1 \text{ 时} \end{cases} \tag{7-100}$$

$$S'_G = \begin{cases} 0 & \text{当 } S_{CL} = 0 \text{ 时} \\ 1 & \text{当 } S_{CL} = 1 \text{ 时} \end{cases} \tag{7-101}$$

7.3.2.3 进程互配结构模型

进程互配结构模型是按进程互配仿真策略组建的。进程互配仿真策略认为模型的主动成分各项活动是相互关联的，组成一个相互关联而不是互不相关的事件。这个相关组合称为进程。一个成分进入进程后，一般将进行到进程的全部活动结束。因此，进程互配仿真策略是以模型的各主动成分的活动为主线来调度事件生成顺序的。由于仿真程序与模型结构间对应关系明确，因此，编程较方便，首次成功率高。

其具体做法是：把上节讨论的各主动成分的局部状态转移函数分解为若干子转移函数，每个子转移函数与主动成分的一个控制状态相对应。控制状态负责相应子转移函数的执行。对应于各子转移函数，条件函数也分解成子条件函数。为此，在进程互配结构模型中，主动成分用二维数组 (l_a, t_a) 描述。其中，l_a 是 a 成分的控制状态。t_a 是下一生成时刻。若 l_a 的值域为 $\{0,1,\cdots,M\}$，则 l_a 相对应的子条件函数 $C_{l_a}(s)$ 与子条件函数 $\delta_{l_a}(s)$ 分别为：$C_{0a}(s),\cdots C_{Ma}(s)$ 及 $\delta_{0a}(s),\cdots,\delta_{Ma}(s)$。而各子转移函数对应于一个进程，若给各子转移函数程序块以相应的功能名称，则进程互配模型可直观地了解仿真活动的内容。

下面还以计算机和用户终端系统为例说明有关步骤。

(1) 组成成分

组成成分与活动扫描结构模型相同。

(2) 描述变量

①描述用户 i 的变量 $(i = 1,2\cdots,90)$

用户 i 的控制状态 l_i 值域 $\{0,1,2,3,4\}$。

用户 i 的下一生成时刻 t_i，值域 $\{R^+_{0,\infty}\}$。

②描述工作人员的变量

工作人员控制状态 l_{CL}，值域 $\{0,1\}$。

工作人员下一生成时刻 t_{CL}，值域 $\{R^+_{0,\infty}\}$。

③描述计算机的变量

计算机状态 S_{CPU}，值域 $\{0,1\}$。

④描述用户队的变量

用户队状态 S_{UQ}，值域 $\{0,1,2,\cdots,90\}^*$。

⑤描述批次队的变量

批次队状态 S_{BQ}，值域 $\{0,1,\cdots,90\}$ 的幂集合。

⑥描述批次队门的变量

批次队门状态 S_G，值域 $\{0,1\}$。

(3) 参数

用户类型函数 TYPE(i)。

(4) 相互关系

用户 i 的状态：

0——预定生成时刻 $t_i = t + \mathrm{WCT}$，将 l_i 置 1。

1——用户队空且计算机空闲时转 2，否则进用户队，并将控制状态 l_i 置 2。

2——预定生成时刻 $t_i = t + ST$，将 l_i 置 3。

3——离开用户队，进入批次队，l_i 置 4。

4——出门。l_i 转初值 0 状态。

工作人员状态：

0——打开门，转入控制 1 状态。即 $l_{CL} = 1$。

1——关门，预定生成时间 $t_{CL} = t + 8 \times 60$，转入控制 0 状态。

计算机状态：

0——空闲。1——忙。

批次队门状态：

0——开。1——关。

用户 i 的子条件函数和子转移函数如表 7－3。

表 7－3　用户 i 的子条件函数和子转移函数

l	$C_{li}(s)$	$\delta_{li}(s)$
0	≡true	$l'_i = 1$ $t'_i = t_i + WCT(\text{TYPE}(i), r_{i,1})$ $S'_{UQ} = S_{UQ}$ $S'_{BQ} = S_{BQ}$ $r'_{i,1} = \Gamma(r_{i,1})$ $r'_{i,2} = r_{i,2}$
1	≡true	转控制状态 2，执行 $\delta_{2i}(s)$ 当 $S_{UQ} = A$ 且 $S_{CPU} = 0$ 时 其余情况：$l'_i = 2$；$t'_i = t_i$；$S'_{UQ} = S_{UQ}$；$S'_{BQ} = S_{BQ}$；$r'_{i,1} = r_{i,1}$；$r'_{i,2} = r_{i,2}$
2	≡（$i = \text{top}(S_{UQ})$ 且 $S_{CPU} = 0$）	$l'_i = 3$ $t'_i = t_i + ST(\text{TYPE}(i), r_{i,2})$ $S'_{UQ} = S_{UQ}$ $S'_{BQ} = S_{BQ}$ $r_{i,1} = r_{i,1}$ $r'_{i,2} = \Gamma(r_{i,2})$
3	≡true	$l'_i = 4$ $t'_i = t_i$ $S'_{UQ} = \text{rest}(S_{UQ})$ $S'_{BQ} = S_{BQ} \cup \{i\}$ $r'_{i,1} = r_{i,1}$ $r'_{i,2} = r_{i,2}$

续表

l	$C_{li}(s)$	$\delta_{li}(s)$
4	$\equiv (S_G=0)$	$t'_i=t_i$ $S'_{UQ}=S_{UQ}$ $S'_{BQ}=S_{BQ}-\|i\|$ $r'_{i,1}=r_{i,1}$ $r'_{i,2}=r_{i,2}$ 转控制状态0

表7-4 工作人员的子条件函数和子转移函数

l	$C_{lCL}(s)$	$\delta_{lCL}(s)$
0	≡true	$S'_G=1$ $t'_{CL}=t_{CL}$ 转控制状态1。
1	$\equiv (S_{BQ}=A)$	$S'_G=0$ $l'_{CL}=0$ $t'_{CL}=t_{CL}+8\times 60$

表7-4是工作人员的子条件函数和子转移函数。由于$\delta_{l1}(s),\cdots,\delta_{l90}(s)$各子函数的结构是相同的，因此，编程时用$\delta_l(i,s)$描述，其中$i$是用户参数。

7.3.3 离散事件系统仿真结果分析

由于离散事件系统的仿真模型中包含随机变量，因此，每次运行得到的只是大量抽样数据的统计结果。它与总体统计特性之间存在的误差也是随机变量，与仿真运行方式、结果处理方法等有关。因此，在离散事件系统仿真后，必须对仿真结果进行数理统计分析。而一般的统计方法是假设所有测量值相互独立，并要求总体分布是稳定的，即总体分布不受抽样次数影响，不随时间变化，同时一次抽样值不受其他抽样值的任何方式的影响。离散事件系统仿真所收集到的观测值大多数不满足上述条件，为提高仿真结果的可信度，必须采用一些必要的措施。

7.3.3.1 复演法

复演法是通过重复运行仿真，得到相互独立的样本序列，从而确定仿真结果的置信区间。重复运行仿真是选用不同的独立随机数序列，对系统在相同参数、初始条件、抽样次数等条件下重复运行，获得相互独立的抽样样本序列。

若重复运行p次，令x_{ij}是第j次运行中第i次观测值。如第j次运行的样本均值为$\overline{x_j}(n)$，方差为$\sigma_j^2(n)$。则p次运行结果，得到的是相互独立的同一分布的随机变量序列$\{X_j\}, j=1,2,\cdots,p$。用$\overline{X}$表示容量为p的样本$\{\overline{X_j}\}$的均值，σ^2为样本的方差估计。则有：

$$\overline{X}=\frac{1}{p}\sum_{j=1}^{p}\overline{X}_j$$

$$\sigma^2=\frac{1}{p-1}\sum_{j=1}^{p}(\overline{X}_j-\overline{X})^2$$

可以证明统计量

$$t_{p-1} = \frac{\overline{X} - u}{\sigma/\sqrt{p}} \tag{7-102}$$

服从自由度 $p-1$ 的 t 分布。

给定显著水平 α，根据 t 分布表，可以确定总体均值 μ 的置信区间为：

$$\left[\overline{X} - \sigma/\sqrt{p} \cdot t_{p-1}, \frac{\alpha}{2}, \overline{X} + \sigma/\sqrt{p} \cdot t_{p-1}, \alpha/2\right] \tag{7-103}$$

置信度为 $100(1-\alpha)\%$ 。

重复次数 p 与每次仿真的抽样数 n 应选择合适。为提高置信度，在相同运行机时条件下，应减小重复次数 p，加大抽样数 n（pn 保持不变）。而 p 值越大，样本均值才能服从正态分布。

7.3.3.2 批均值法

把一次长时间仿真运行分为若干段，每段为一批，用相同的抽样数，然后对每批样本求均值。这种方法称批均值法。这些样本均值可近似为相互独立的同一分布随机变量。由于每批运行结果的状态作为下一批运行的初始状态，因此比用重复运行时把空载状态作为初始状态更符合实际。但也因此引入了批与批之间的相关性。

为了使各批的均值互相独立，对批运行时间应选择足够长，即批容量 n 应足够大，这样，前一部分数据对后一部分数据的影响就会随着数据间距离的增加而减弱。

与复演法相类似，批均值法中的分批数和批容量间也要统筹兼顾。

对比实验表明，批均值法优于复演法。

7.3.3.3 再生法

在系统仿真时，随着仿真时钟的推进，系统的状态变量数值不断变化。如果系统在某一时刻的一组状态变量数值在若干时间后重复产生，这种系统称为再生系统。再生系统中选择一组特殊的状态变量数值作为参考点。把重新产生参考点的时间点称为再生点。系统中一个再生点到下一个再生点巡回（tour）一次，称为一个巡回期。可以看到，巡回期和再生点是随机的。再生法的基本思想就是找出仿真过程中的这些再生点，在再生点时，认为系统重新开始运行，即系统再生。从每个再生点开始的巡回期中获得的统计样本可认为是独立的同一分布的。因此，可用经典统计方法进行参数估计和确定参数的置信区间。

例如对于一个具有泊松到达模式和指数分布的服务时间的单服务系统，采用先入先出排队规则。若系统从空闲状态开始，当第一实体到达时，系统转入忙状态，此期间第二实体到达并进入队列排队，到第一实体服务完毕，按先入先出分别对队首的实体进行服务。直到队列空时，系统转入空闲。可以看到，系统是以一个忙期和一个闲期依次循环的。把系统由闲转忙的时间点作为系统的再生点，此时系统再生，开始一次巡回。

不考虑巡回时，总观测次数为 N，每个实体等待时间为 w_1、w_2、…、w_N，则实体平均等待时间的估计值是：

$$\overline{W} = \frac{1}{N}\sum_{i=1}^{N} w_i \tag{7-104}$$

当按巡回分组时，令 Y_j 是 j 次巡回中各实体等待时间总和，n_j 是 j 次巡回中受到服务的实体数，共有 p 次巡回，则平均等待时间估计值可表示为：

$$\overline{W} = \frac{Y_1 + Y_2 + \cdots + Y_p}{n_1 + n_2 + \cdots + n_p} \tag{7-105}$$

如用 $\overline{Y}$ 表示一次巡回中各实体等待时间总和的估计，$\overline{n}$ 表示相应的受到服务实体数的估计，则有：

$$\overline{Y} = \frac{1}{p}\sum_{j=1}^{p} Y_j ; \overline{n} = \frac{1}{p}\sum_{j=1}^{p} n_j$$

因此，有

$$\overline{W} = \frac{\overline{Y}}{\overline{n}} \tag{7-106}$$

由于实体等待时间是一个随机变量，为此定义 V_j：

$$V_j = Y_j - E(W)n_j \tag{7-107}$$

它是独立的同一分布的随机变量，均值为零，因此，根据中心极限定理，可以求得平均等待时间估计值的置信区间是：

$$\overline{W} \pm \frac{\sigma u_{\alpha/2}}{n\sqrt{p}} \tag{7-108}$$

式中，$u_{\alpha/2}$是对应显著水平为 α 的标准正态分布的临界限，σ^2 是 V_j 的方差。

考虑到 Y_j 与 n_j 并不相互独立，而是有较强的正相关特性，可以求其协方差 S_{12}^2及 Y_j、n_j 的样本方差S_{11}^2及 S_{22}^2。则 V_j 的样本方差S_v^2 可由

$$S_v^2 = S_{11}^2 - 2\overline{W}S_{12}^2 + \overline{W}^2 S_{22}^2 \tag{7-109}$$

求得。而且用S_v^2代替未知的方差σ^2，即可求得再生法时得到的平均等待时间估计值置信区间。

再生法不是对所有模型的所有输出都能适用。但对多数模型的如所有逗留时间、等待时间、队长等输出结果是适用的。

7.3.4 方差减小技术

为了分析和正确使用仿真结果，需应用统计技术处理仿真输出数据。而一个普通的统计分析常需很高的费用。为使相同仿真次数内获得较小方差的仿真输出的方法，称为方差减小技术（Variance Reducing Technique）。常用的方差减小技术有重要性法和相关法。相关法包括相关采样法，对偶变量法和控制变量法等。

7.3.4.1 相关采样法

相关采样法又称公用随机数法。它是在整个仿真时间内，用相同的单位均匀分布种子随机数流驱动几个不同的模型，这样所观测到的差别是由系统设计本身的差别所造成而不是由于采用不同随机数流而造成的。相关采样法适用于具有正相关的系统模型。当要对比的模型数多于两个时，由于统计分析要引入相关性而使计算复杂化。

7.3.4.2 对偶变量法

对偶变量法适用于单个系统的仿真，它把系统的仿真分为对偶的两段，使这两段的观测值形成负相关，从而使两段的观测值互相补偿，而其平均值的方差得到减小。具体实施的方法是采用互补的随机数驱动一个系统的两段，从而形成负相关。即一个随机数序列为 u_k，则另一个采用 $1-u_k$。

对偶变量法与相关采样法类似，它也依赖于模型，而且对于方差减小的程度不能事先知道，此外，要在编程时，注意随机数序列的同步。

7.3.4.3 控制变量法

和相关采样法及对偶变量法一样，控制变量法设法利用某些随机变量的相关性来减小方差的。例如为了减小随机变量 X 的方差，找一个控制变量 Y，它也是一个随机变量，但与 X 是相关的。因此，从数学期望已知的意义上讲，Y 控制 X。由于 Y 的数学期望已知，则 $X_c = X - a(Y - \gamma)$ 对任意实数 a 是 $\mu = E(x)$ 的一个无偏估计。式中，$\gamma = E(Y)$，a 是权。

根据统计分析知识，有：

$$D[X_c] = D[X] + a^2 D[Y] - 2a\mathrm{Cov}(X, Y) \tag{7-110}$$

式中，$\mathrm{Cov}(X, Y)$ 是 X, Y 的协方差。因此，当

$$2a\mathrm{Cov}(X, Y) > a^2 D[Y]$$

时，$D[X_c] < D[X]$，即 X_c 的方差比 X 的方差要小。本方法的关键是能找到好的控制变量及选择合适的权 a。

7.3.5 通用离散事件系统仿真语言

50 年代末以来，国外出现了很多仿真语言，GPSS 是一种适用于确定性系统和随机性系统的离散事件系统仿真语言。它的使用简单、通用性较强，而以 FORTRAN 语言编写的 GPSS－F77 由于易于移植而得到广泛应用。

GPSS 采用进程互配结构，包括流程管理、模块集及支撑程序等三大部分。流程管理包括对模型各种状态变化的推进时序的控制。模块集则是子转移函数等全部模块的集合。支撑程序包括预置、打印、收集统计等内部子程序。

用户编写仿真语言程序时，只要把为说明所建模型的语句（大多数为调用语句）写入由 GPSS－F77 提供的框架的规定位置即可。按功能，框架分为 12 段。用户只需填写部分内容。各段功能如下：

①通用 FORTRAN 说明段

用户需填入所用变量的全部数组维数等必要说明。

②语句函数段

若用户使用语句函数，则在此段定义。

③数据区清零段

用 RESET 子程序对全部数据区清零。

④初始化段

用户在此段初始化常量和控制变量。

⑤读入数据段

为使用户便于使用和修改、在执行程序时，用 ITREAD 函数读入控制值、初值和功能值表。

⑥预定初始事件段

为使仿真能运行，用调用 EVENT 的方法预定一些初始事件。

⑦流程管理 1：事件激发和预定激发段

用 ACTIV1 找出最早事件或预定事件激发，把它们在事件表中的项清除。

⑧标号选择语句段

用户填写 GOTO 语句，使控制跳转。

⑨模型段

用户在此段写出仿真的模型。即调用 GPSS－F77 子程序的有关语句。

⑩流程管理 2：条件激发段

用 ACTIV2 检测有否新的条件激发并进行相应流程管理。

⑪最终分析段

调用 ENDBIN 计算收集在 BIN 中的最终结果。也可用户编写分析仿真程序运行结果的程序段。

⑫打印结果段

用 REPORT 打印出全部数据区信息。

有关 GPSS－F77 的程序和使用方法可参见有关资料。

离散事件系统仿真语言除了 GPSS－F77 外，常用的还有 SIMSCRIPT、SIMULA 等。

参 考 文 献

1. Geoffercy Gordon, System Simulation, Prentice－Hall, Inc., New Jersey, 1978

2. 熊光楞，控制系统数字仿真，清华大学出版社，1982

3. 熊光楞，数字仿真算法与软件，宇航出版社，1991

4. 任鑫权，控制系统计算机仿真，机械工业出版社，1988

5. 涂　健，控制系统的数字仿真与计算机辅助设计，华中工学院出版社，1985

6. 孙增圻等，控制系统的计算机辅助设计程序汇编，清华大学出版社，1988

7. 南京大学，常微分方程的数值解法，科学出版社，1979

8. 张德荣等，计算方法与算法语言，人民出版社，1981

9. B. P. Zeigler, Theory of Modelling and Simulation, John Wiley & Sons, New York, 1976

10. J. R. Ragazzini & G. F. Franklin, Sampled Data Control Systems, Mcgrawhill, New York, 1960

11. R. G. Jacquot, Modern Digital Control Systems, Marcel Dekker, Inc., 1981

12. 蒋慰孙，系统控制，华东化工学院出版社，1988

13. B. C. Kuo, Automatic Control Systems, Prentice－Hall, Inc., 1975

14. 绪方胜彦，卢伯英译，现代控制工程，科学出版社，1978

15. E. E. Mitchell & C. F. Olsen, Digital Computation of Frequency Response, Simulation, Vol. 25, No. 5, 1975

16. H. H. Rosenbrock, Computer－aided Control System Design, Acadamic Press, 1974

17. M. A. Johnson, Diagonal dominance and the method of pseudodiagonalisation, Proc. IEE, Vol. 119, 1972

18. 白方周、李嗣福，多变量系统逆奈氏阵列法设计程序，中国科技大学，1982

19. 范鸣玉、张　莹，最优化技术基础，清华大学出版社，1982

20. 宋承龄、王章德，系统仿真，国防工业出版社，1989